Regularized Large Neighborhood Search

Germain Vivier-Ardisson
Google DeepMind & CERMICS
Paris, France
gvivier@google.com

Laurent Demonet
Google Research
Paris, France
demonet@google.com

Axel Parmentier
CERMICS, ENPC, CNRS, IPP
Marne-la-Vallée, France
axel.parmentier@enpc.fr

Mathieu Blondel
Google DeepMind
Paris, France
mblondel@google.com

Abstract

Operations research practitioners typically tackle NP-hard combinatorial problems using large neighborhood search (LNS), a scalable heuristic that iteratively refines a current solution by locally re-optimizing subsets of its variables. In contrast, most existing approaches for integrating combinatorial optimization layers into neural networks still assume access to an exact global solution, which is computationally intractable. We bridge this gap by introducing regularized LNS (RLNS). By regularizing or perturbing local subproblems, we turn the LNS heuristic into an efficient MCMC sampler over the combinatorial set of feasible solutions, with associated Fenchel-Young losses. Under entropic regularization, we prove that RLNS performs exact block Gibbs sampling. Furthermore, adjusting the number of RLNS iterations allows us to interpolate between pseudolikelihood and exact maximum likelihood estimation, for end-to-end learning without global solvers. We demonstrate our approach on k-subset selection, generalized assignment, and stochastic vehicle scheduling problems.

1 Introduction

Integrating combinatorial optimization solvers into neural networks has emerged as a powerful paradigm for structured prediction [Dalle et al., 2022, Mandi et al., 2024, Sadana et al., 2025, Schiffer et al., 2026]. Such architectures enable machine learning models to make discrete decisions that satisfy complex constraints, via output layers of the form

$$\boldsymbol{\theta} \longmapsto \hat{\boldsymbol{y}}(\boldsymbol{\theta}) \triangleq \operatorname*{arg\,max}_{\boldsymbol{y} \in \mathcal{Y}} \langle \boldsymbol{\theta}, \boldsymbol{y} \rangle, \tag{1}$$

where $\mathcal{Y}$ is a finite, but combinatorially large, set of feasible outputs. The fundamental challenge in this setting is differentiability. Because the $\arg\max$ operator provides zero gradients almost everywhere, standard approaches typically rely on introducing a regularization term to yield a continuous, differentiable relaxation [Niculae et al., 2018, Mandi and Guns, 2020, Berthet et al., 2020]. As summarized in Table 1, various methods have been proposed to solve the resulting problem, depending on the type of regularization used and available oracle. However, most of the existing literature fundamentally assumes access to a *global* exact solver for Problem (1) (a global MAP oracle) or marginal oracle [Lafferty et al., 2001, Wainwright and Jordan, 2008], which computes the expectation of the Gibbs distribution $\boldsymbol{p}(\boldsymbol{y}) \propto \exp(\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle)$ over the entire combinatorial set $\mathcal{Y}$. For NP-hard optimization problems encountered in operations research, calling a global oracle at every training step can be computationally intractable.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Table 1: The new connection we establish between LNS and Gibbs sampling enables us to leverage local oracles and effectively regularize the problem when usual global oracles are not available.

	Regularization	**Oracle**	**Approach**
Differentiable DP (2018)	Entropic	Global marginal	DP
SparseMAP (2018)	Quadratic	Global MAP	Frank-Wolfe
Barrier FW (2015)	TRW Entropy	Global MAP	Frank-Wolfe
IntOpt (2020)	Logarithmic barrier	Interior point solver	Primal-dual
Perturbed optimizers (2020)	Implicit via noise	Global MAP	Monte-Carlo
DYS-net (2024)	Quadratic	Projection oracles	Davis-Yin splitting
Blackbox solvers (2020)	None	Global MAP	Interpolation
Local search MCMC (2026a)	Entropic	Local search	Metropolis-Hastings
RLNS (proposed)	Entropic / Implicit via noise	*Local* marginal / MAP	Gibbs sampling

Instead, practitioners rely on highly scalable heuristics, particularly large neighborhood search (LNS). LNS iteratively improves a feasible solution by freezing a subset of its variables and exactly re-optimizing the remaining free variables [Pisinger and Ropke, 2018], thereby relying on *local* MAP oracles only. Our goal is not to improve LNS as an optimization heuristic, but to leverage these local oracles to enable learning in combinatorial spaces where global solvers fail. To bridge this gap, we frame gradient estimation as a sampling problem and regularize LNS into a rejection-free MCMC sampler over the feasible set. While recent work connected local search heuristics with Metropolis-Hastings [Vivier-Ardisson et al., 2026a], our approach connects LNS to Gibbs sampling. Specifically, we make the following contributions:

- In Section 3.1, we introduce regularized LNS (RLNS), a stochastic version of LNS based on smoothed maximum operators, integrating regularization into the local subproblems.
- In Section 3.2, we prove that under entropic regularization, RLNS exactly implements a block Gibbs sampler whose stationary distribution matches a globally regularized Gibbs distribution.
- In Section 3.3, we show that RLNS supports perturbation-based regularization [Berthet et al., 2020], leading to tractable updates using the standard local MAP oracles of LNS heuristics.
- In Section 4, we demonstrate that the number of RLNS iterations allows us to interpolate between pseudolikelihood [Besag, 1974] and exact maximum likelihood estimation.
- In Section 5, we evaluate RLNS on a controlled estimation task on κ-hot vectors, a generalized assignment problem learning task, and a stochastic vehicle scheduling problem.

A summary of notations is available at the beginning of the appendix.

2 Background and related work

Large neighborhood search. LNS is a meta-heuristic used to approximately solve Problem (1) when it is intractable (e.g., when it is NP-hard), by using a *local* solver (*local* MAP oracle):

$$\boldsymbol{\theta} \mapsto \operatorname*{arg\,max}_{\boldsymbol{y}' \in \mathcal{Y}} \langle \boldsymbol{\theta}, \boldsymbol{y}' \rangle \quad \text{s.t.} \quad \boldsymbol{y}'_{[S]} = \boldsymbol{y}_{[S]}. \tag{2}$$

Such an oracle exactly solves sub-problems of (1) by fixing a subset $S \subseteq [d]$ of variables to the previous solution $\boldsymbol{y} \in \mathcal{Y}$. LNS iteratively refines a solution by freezing different variables at each step and exactly re-optimizing the rest [Archetti and Speranza, 2014, Pisinger and Ropke, 2018]. The pseudo-code is given in Algorithm 1.

Algorithm 1 Large neighborhood search (LNS)

Inputs: Logits $\boldsymbol{\theta} \in \mathbb{R}^d$, feasible set $\mathcal{Y} \subset \mathbb{R}^d$, initial solution $\boldsymbol{y}^{(0)} \in \mathcal{Y}$, number of iterations $K \in \mathbb{N}$.
for $k = 0$ to $K - 1$ **do**
  Choose a subset of frozen coordinates $S^{(k)} \subset [d]$.
  Define the local neighborhood: $\mathcal{Y}^{(k)} \triangleq \left\{ \boldsymbol{y} \in \mathcal{Y} \mid \boldsymbol{y}_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]} \right\}$
  Set $\boldsymbol{y}^{(k+1)} \leftarrow \operatorname{arg\,max}_{\boldsymbol{y} \in \mathcal{Y}^{(k)}} \langle \boldsymbol{\theta}, \boldsymbol{y} \rangle$ {Solve the local optimization subproblem.}
end for
Return $\boldsymbol{y}^{(K)}$

Smoothed maximum operators. For a finite set $\mathcal{Z}$, let $\Omega : \triangle^{\mathcal{Z}} \to \mathbb{R}$ be a lower semicontinuous (l.s.c.) strictly convex regularizer. The smoothed maximum operator $f_\Omega : \mathbb{R}^{|\mathcal{Z}|} \to \mathbb{R}$ [Mensch and Blondel, 2018] and smoothed maximizer $\boldsymbol{p}_\Omega : \mathbb{R}^{|\mathcal{Z}|} \to \triangle^{\mathcal{Z}}$ are defined as:

$$f_\Omega(\boldsymbol{s}) \triangleq \max_{\boldsymbol{q} \in \triangle^{\mathcal{Z}}} \langle \boldsymbol{s}, \boldsymbol{q} \rangle - \Omega(\boldsymbol{q}) = \Omega^*(\boldsymbol{s}), \quad \text{and} \quad \boldsymbol{p}_\Omega(\boldsymbol{s}) \triangleq \operatorname*{arg\,max}_{\boldsymbol{q} \in \triangle^{\mathcal{Z}}} \langle \boldsymbol{s}, \boldsymbol{q} \rangle - \Omega(\boldsymbol{q}) = \nabla \Omega^*(\boldsymbol{s}),$$

where $\cdot^*$ denotes the convex conjugate and the rightmost equality is given by Danskin's theorem [Danskin, 1966]. When Ω is *Legendre-type* [Rockafellar, 1970], i.e., strictly convex and differentiable on $\operatorname{ri}(\triangle^{\mathcal{Z}})$ with gradient norm diverging at the relative boundary of $\triangle^{\mathcal{Z}}$, we have $\operatorname{range}(\boldsymbol{p}_\Omega) = \operatorname{ri}(\triangle^{\mathcal{Z}})$. Typical choices for Ω include Shannon's negative entropy, yielding $f_\Omega(\boldsymbol{s}) = \log \sum_{\boldsymbol{z} \in \mathcal{Z}} \exp(s_{\boldsymbol{z}})$ and the *softmax* $\boldsymbol{p}_\Omega(\boldsymbol{s})_{\boldsymbol{z}} = \exp(s_{\boldsymbol{z}}) / \sum_{\boldsymbol{z}'} \exp(s_{\boldsymbol{z}'})$, or the squared Euclidean norm $\frac{1}{2}\|\cdot\|_2^2$, giving the *sparsemax* operator instead [Martins and Astudillo, 2016]. In Section 3.3, we define Ω via low-rank dual additive noise (Definition 4). Like Shannon's negative entropy (and unlike $\frac{1}{2}\|\cdot\|_2^2$), it is Legendre-type when $\mathcal{Z} \subset \mathbb{R}^d$ consists of extreme points of $\operatorname{conv}(\mathcal{Z})$ [Bouvier et al., 2025].

Global Fenchel-Young learning. Following Blondel et al. [2020], Problem (1) can be lifted over the distribution simplex $\triangle^{\mathcal{Y}}$ using the *score vector* $\boldsymbol{s}_{\boldsymbol{\theta}} \triangleq (\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle)_{\boldsymbol{y} \in \mathcal{Y}} \in \mathbb{R}^{|\mathcal{Y}|}$. This enables adding distribution-space regularization into the problem via a *global* regularizer $\Omega : \triangle^{\mathcal{Y}} \to \mathbb{R}$.

Definition 1 (Distribution-space Fenchel-Young loss)**.** *For a l.s.c. strictly convex regularizer $\Omega : \triangle^{\mathcal{Y}} \to \mathbb{R}$, the* distribution-space Fenchel-Young loss $L_\Omega : \mathbb{R}^d \times \triangle^{\mathcal{Y}} \to \mathbb{R}_+$ *is defined as:*

$$L_\Omega(\boldsymbol{\theta}\,; \boldsymbol{p}) \triangleq \Omega^*(\boldsymbol{s}_{\boldsymbol{\theta}}) + \Omega(\boldsymbol{p}) - \langle \boldsymbol{s}_{\boldsymbol{\theta}}, \boldsymbol{p} \rangle = f_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}}) + \Omega(\boldsymbol{p}) - \langle \boldsymbol{\theta}, \mathbb{E}_{Y \sim \boldsymbol{p}}[Y] \rangle.$$

This loss is convex in $\boldsymbol{\theta}$, smooth if Ω is strongly convex, non-negative, and is zero when $\boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}}) = \boldsymbol{p}$. Minimizing L_Ω uses the gradient $\nabla_{\boldsymbol{\theta}} L_\Omega(\boldsymbol{\theta}\,; \boldsymbol{p}) = \mathbb{E}_{\boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})}[Y] - \mathbb{E}_{\boldsymbol{p}}[Y]$, requiring a global oracle over $\mathcal{Y}$: a marginal oracle in the entropic case ($\boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})(\boldsymbol{y}) \propto \exp(\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle)$), or a MAP oracle to estimate $\mathbb{E}_{Z \sim \mathcal{D}}[\arg\max_{\boldsymbol{y} \in \mathcal{Y}} \langle \boldsymbol{\theta} + \varepsilon Z, \boldsymbol{y} \rangle]$ under perturbation-based regularization (Section 3.3). Evaluating such global oracles at every step is intractable for NP-hard problems.

Differentiating through combinatorial optimization. Most approaches to differentiate Problem (1) with respect to $\boldsymbol{\theta}$ introduce regularization tailored to the available oracle. If the original problem admits a dynamic programming solution, one can solve an entropy-regularized version via algorithmic smoothing [Li and Eisner, 2009, Mensch and Blondel, 2018]. For linear programs, interior-point solvers can compute a log-barrier-regularized solution [Mandi and Guns, 2020], while projection oracles can be used for quadratic regularization [McKenzie et al., 2024]. Global MAP oracles can also be used to solve a regularized problem, either via Frank-Wolfe [Niculae et al., 2018, Krishnan et al., 2015] or via Monte-Carlo estimation with input perturbations [Berthet et al., 2020, Dalle et al., 2022]. Regarding differentiation, several strategies are possible. When the approach only needs to differentiate through a (regularized) $\max$, as for Fenchel-Young losses [Blondel et al., 2020], we can use Danskin's theorem [Danskin, 1966]. When differentiating a (regularized) $\arg\max$, one can implicitly differentiate KKT conditions [Agrawal et al., 2019, Ferber et al., 2020, Blondel et al., 2022], or compute surrogate gradients via interpolation of the solver [Vlastelica et al., 2020].

Local oracles in structured prediction. MAP inference in graphical models [Wainwright and Jordan, 2008] is inherently a combinatorial optimization problem. In both domains, the intractability of global inference motivates the use of local oracles. For example, the seminal iterated conditional modes algorithm [Besag, 1986] deterministically maximizes local conditional probabilities, acting as the graphical models analogue to (unregularized) LNS. During learning, intractable global marginal inference is typically bypassed using local oracles: pseudolikelihood (or composite likelihood) methods [Besag, 1974, Lindsay, 1988, Varin et al., 2011] optimize local conditionals of observed data, while local perturb-and-MAP [Bertasius et al., 2017] forms their noise-regularized counterpart. Just like contrastive divergences [Hinton, 2002], which interpolate between pseudolikelihood (CD-1) and maximum likelihood estimation (CD-∞) [Hyvärinen, 2006, Asuncion et al., 2010], we will show that the number of RLNS iterations enables interpolating between local (RLNS-1) and global Fenchel-Young learning (RLNS-∞). While Berthet et al. [2020] successfully brought global perturb-and-MAP learning [Papandreou and Yuille, 2011] to combinatorial layers, leveraging scalable local oracles remains a critical gap in the field which our work aims to bridge.

3 Regularized LNS

3.1 Proposed framework

We now introduce our main contribution: using smoothed maximum operators, we define a stochastic, regularized version of LNS (RLNS), whose iterates follow a Markov chain with state space $\mathcal{Y}$. To do so, we first define a family of local regularization functions from a global one, via inclusion maps.

Definition 2 (Local regularizers)**.** *For any subset of solutions $\mathcal{Z} \subseteq \mathcal{Y}$, we define the inclusion map $\iota_{\mathcal{Z}} : \triangle^{\mathcal{Z}} \hookrightarrow \triangle^{\mathcal{Y}}$ from local distributions on $\mathcal{Z}$ to global distributions on $\mathcal{Y}$ via padding:*

$$\iota_{\mathcal{Z}}(\boldsymbol{q}) \triangleq \begin{pmatrix} \begin{cases} q_{\boldsymbol{y}} & \text{if } \boldsymbol{y} \in \mathcal{Z} \\ 0 & \text{else} \end{cases} \end{pmatrix}_{\boldsymbol{y} \in \mathcal{Y}}.$$

Then, for any lower semicontinuous, strictly convex global regularizer $\Omega : \triangle^{\mathcal{Y}} \to \mathbb{R}$, we define its local counterpart $\Omega^{\mathcal{Z}} : \triangle^{\mathcal{Z}} \to \mathbb{R}$ by composing it with $\iota_{\mathcal{Z}}$, as $\Omega^{\mathcal{Z}} \triangleq \Omega \circ \iota_{\mathcal{Z}}$.

While standard approaches given in Table 1 aim to solve a regularized version of the global optimization problem (1), we propose to regularize the LNS subproblems from Algorithm 1 using the local regularizer $\Omega^{\mathcal{Y}^{(k)}}$, to define local distributions $\boldsymbol{p}_{\Omega^{\mathcal{Y}^{(k)}}}\left((\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]}\right) \in \triangle^{\mathcal{Y}^{(k)}}$. RLNS is then obtained by iteratively sampling from these locally regularized distributions, as shown in Algorithm 2.

Algorithm 2 Regularized LNS with K iterations (RLNS-K)

1: **Inputs:** Logits $\boldsymbol{\theta}$, feasible set $\mathcal{Y}$, initial solution $\boldsymbol{y}^{(0)}$, number of iterations K, regularization Ω.
2: **for** $k = 0$ to $K - 1$ **do**
3: Choose a subset of frozen coordinates $S^{(k)} \subset [d]$.
4: Define the local neighborhood: $\mathcal{Y}^{(k)} \triangleq \left\{ \boldsymbol{y} \in \mathcal{Y} \mid \boldsymbol{y}_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]} \right\}$
5: Sample $\boldsymbol{y}^{(k+1)} \sim \boldsymbol{p}_{\Omega^{\mathcal{Y}^{(k)}}}\left((\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]}\right)$ {Sample from the local regularized distribution.}
6: **end for**
7: **Return** $\frac{1}{K}\sum_{k=1}^{K} \boldsymbol{y}^{(k)}$

Ergodicity. To guarantee that the distribution of the iterates $\boldsymbol{y}^{(k)}$ of Algorithm 2 asymptotically converges to a *unique* stationary distribution independent of the initialization $\boldsymbol{y}^{(0)}$ (i.e., that the Markov chain is *ergodic*), the two following properties are sufficient:

1. The algorithm employs an *irreducible variable selection scheme*, i.e., the distribution over frozen variables $S^{(k)}$ induces a connected transition graph over the feasible set $\mathcal{Y}$.
2. The local update distributions are *dense*, i.e., $\boldsymbol{p}_{\Omega^{\mathcal{Y}^{(k)}}}\left((\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]}\right)(\boldsymbol{y}) > 0 \; \forall \boldsymbol{y} \in \mathcal{Y}^{(k)}$.

The first condition ensures that LNS can reach any feasible solution from any other one. This requirement is not very stringent in practice, and is typically met by practical implementations which either consider reduced scope or exploit decomposition into subproblems with specific structure [Archetti and Speranza, 2014]. RLNS can therefore directly leverage existing practitioner knowledge and domain-specific variable selection heuristics from standard LNS.

The second condition (satisfied, e.g., when $\Omega^{\mathcal{Y}^{(k)}}$ is Legendre-type) ensures positive path probabilities and thus irreducibility. Since $\boldsymbol{y}^{(k)} \in \mathcal{Y}^{(k)}$, dense updates also yield positive self-loop probabilities, ensuring aperiodicity and therefore ergodicity [Levin et al., 2008].

Definition 3 (Stationary distribution of RLNS)**.** *When the Markov chain $(\boldsymbol{y}^{(k)})_{k \in \mathbb{N}}$ defined by Algorithm 2 is ergodic, we denote by $\pi^{\mathrm{LNS}}_{\boldsymbol{\theta},\Omega} \in \triangle^{\mathcal{Y}}$ its unique stationary distribution.*

Rejection-free sampling. In contrast to the approach of Vivier-Ardisson et al. [2026a], which turns local search heuristics into Metropolis-Hastings-based MCMC samplers, Algorithm 2 does not use any rejection mechanism, enabling updating more variables at a time without stagnating (see Fig. 1).

3.2 Entropic LNS and Gibbs sampling

We now focus on the case where Ω is Shannon's negative entropy scaled by $\gamma > 0$,

$$\Omega(\boldsymbol{q}) \triangleq -\gamma H^s(\boldsymbol{q}) = \gamma \sum_{\boldsymbol{y} \in \mathcal{Y}} \boldsymbol{q}(\boldsymbol{y}) \log \boldsymbol{q}(\boldsymbol{y}).$$

This choice yields the Gibbs distribution $\boldsymbol{p}_\Omega(\boldsymbol{s})(\boldsymbol{y}) \propto \exp(s_{\boldsymbol{y}}/\gamma)$. Computing its expectation (i.e., performing global marginal inference) is only tractable in certain settings, e.g., if the global problem admits an exact dynamic programming solution [Mensch and Blondel, 2018]. The local RLNS update distribution $\boldsymbol{p}_{\Omega^{\mathcal{Y}^{(k)}}}(\boldsymbol{s}_{[\mathcal{Y}^{(k)}]})$, however, is a *local* Gibbs distribution on $\mathcal{Y}^{(k)}$, and can be sampled from by regularized dynamic programming if the *local* problem can be solved by dynamic programming. This is the case, e.g., of our applications in Sections 5.1 and 5.2.

Proposition 1 (Entropic LNS as a block Gibbs sampler)**.** *Assume $\Omega = -\gamma H^s$ with $\gamma > 0$. For any current solution $\boldsymbol{y}^{(k)} \in \mathcal{Y}$ and frozen subset $S^{(k)} \subseteq [d]$, the locally regularized distribution matches the global regularized distribution conditioned on the fixed variables:*

$$\boldsymbol{p}_{\Omega^{\mathcal{Y}^{(k)}}}\left((\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]}\right)(\boldsymbol{y}) = \boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})\left(\boldsymbol{y} \mid \boldsymbol{y}^{(k)}_{[S^{(k)}]}\right) \quad \forall \boldsymbol{y} \in \mathcal{Y}^{(k)}.$$

*Consequently, Algorithm 2 implements an exact **block Gibbs sampler** with $\boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})$ as a target distribution, and if the variable selection scheme is irreducible, we have:*

$$\pi^{\mathrm{LNS}}_{\boldsymbol{\theta},\Omega}(\boldsymbol{y}) = \boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})(\boldsymbol{y}) \propto \exp\left(\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle / \gamma\right) \quad \forall \boldsymbol{y} \in \mathcal{Y}.$$

The proof is given in Appendix B.1. Thus, in the Shannon-regularized case, the stationary distribution of the locally-regularized iterates of Algorithm 2 matches the globally-regularized distribution. The next proposition shows that $\Omega = -\gamma H^s$ is in fact the only separable regularizer with this property. The proof, given in Appendix B.2, relies on a functional equation-based characterization of Shannon's entropy from information theory [Horibe, 1988, Gselmann, 2011, Mensch and Blondel, 2018].

Proposition 2 (Characterization of stationary exactness)**.** *Let $\Omega : \triangle^{\mathcal{Y}} \to \mathbb{R}$ be a l.s.c. separable Legendre-type regularization function, normalized such that $\Omega(\delta^{\mathcal{Y}}_{\boldsymbol{y}}) = 0$ for any $\boldsymbol{y} \in \mathcal{Y}$. The RLNS stationary distribution matches the global target for all finite feasible sets $\mathcal{Y} \subset \mathbb{R}^d$, all maximization directions $\boldsymbol{\theta} \in \mathbb{R}^d$, and all irreducible variable selection schemes, meaning*

$$\pi^{\mathrm{LNS}}_{\boldsymbol{\theta},\Omega} = \boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}}), \quad \textbf{\textit{if and only if}} \quad \exists \gamma > 0 : \ \Omega = -\gamma H^s.$$

Rao-Blackwellization. When using regularized dynamic programming to sample the next RLNS iterate $\boldsymbol{y}^{(k+1)}$ from the local Gibbs distribution $\boldsymbol{p}_{\Omega^{\mathcal{Y}^{(k)}}}((\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]})$, we can compute its expectation $\mathbb{E}[\boldsymbol{y}^{(k+1)} | \boldsymbol{y}^{(k)}, S^{(k)}]$ with the same complexity. Thus, we can reduce variance by averaging local expectations instead of single samples during MCMC [McKeague and Wefelmeyer, 2000].

3.3 Perturbed LNS and tractable updates

In many applications, a solver for the unregularized local problem in Eq. (2) (local MAP oracle) is available but a solver for the Shannon-regularized local problem (local marginal oracle) is not. To support this setting, we now consider the case of *perturbation-based* regularization, which allows us to make RLNS tractable whenever standard, unregularized LNS is.

Definition 4 (Perturbation-based regularization)**.** *Let $\varepsilon > 0$ be a temperature parameter, and $\mathcal{D}$ be a distribution on $\mathbb{R}^d$ admitting a positive density. Following Bouvier et al. [2025], we define the (global) distribution-space* perturbation-based regularization *function $\Omega_\varepsilon : \triangle^{\mathcal{Y}} \to \mathbb{R}$ as:*

$$\Omega_\varepsilon(\boldsymbol{q}) \triangleq \sup_{\boldsymbol{s} \in \mathbb{R}^{|\mathcal{Y}|}} \left\{ \langle \boldsymbol{s}, \boldsymbol{q} \rangle - \mathbb{E}_{Z \sim \mathcal{D}}\left[\max_{\boldsymbol{y} \in \mathcal{Y}} \{ s_{\boldsymbol{y}} + \varepsilon \langle Z, \boldsymbol{y} \rangle \} \right] \right\}.$$

We show in Appendix B.3 that Ω_ε is strictly convex on $\triangle^{\mathcal{Y}}$. It is also Legendre-type when $\mathcal{Y}$ consists only of extreme points of $\mathrm{conv}(\mathcal{Y})$ [Bouvier et al., 2025]. The corresponding regularized distribution is the *global* perturb-and-MAP distribution [Papandreou and Yuille, 2011, Berthet et al., 2020] $\boldsymbol{p}_{\Omega_\varepsilon}(\boldsymbol{s}_{\boldsymbol{\theta}})(\boldsymbol{y}) = \mathbb{P}_{Z\sim\mathcal{D}}\big(\arg\max_{\boldsymbol{y}'\in\mathcal{Y}}\langle\boldsymbol{\theta}+\varepsilon Z, \boldsymbol{y}'\rangle = \boldsymbol{y}\big)$, and its mean is the expected perturbed *global* maximizer $\mathbb{E}_{\boldsymbol{p}_{\Omega_\varepsilon}(\boldsymbol{s}_{\boldsymbol{\theta}})}[Y] = \mathbb{E}_{Z\sim\mathcal{D}}\big[\arg\max_{\boldsymbol{y}\in\mathcal{Y}}\langle\boldsymbol{\theta}+\varepsilon Z, \boldsymbol{y}\rangle\big]$. We now show that the *local* regularized distributions correspond to *local* perturb-and-MAP distributions.

Proposition 3 (Perturbed LNS sampling)**.** *Let Ω_ε be a perturbation-based regularizer, as defined in Definition 4. Exact sampling from the locally regularized distribution on $\mathcal{Y}^{(k)}$ with local scores $(\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]} \in \mathbb{R}^{|\mathcal{Y}^{(k)}|}$ is tractable via perturbed* local *maximization:*

$$Z \sim \mathcal{D} \implies \boldsymbol{y}^{(k+1)} \triangleq \arg\max_{\boldsymbol{y}\in\mathcal{Y}^{(k)}}\langle\boldsymbol{\theta}+\varepsilon Z, \boldsymbol{y}\rangle \sim \boldsymbol{p}_{\Omega_\varepsilon^{\mathcal{Y}^{(k)}}}\left((\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]}\right).$$

Tractable updates. Proposition 3, proved in Appendix B.4, highlights the computational advantage of using Ω_ε: samples from the local regularized distributions can be obtained by performing perturbed *local* maximization, leveraging the standard LNS oracles from Eq. (2) and Algorithm 1. We summarize this in Algorithm 3, whose iterates match those of Algorithm 2 for $\Omega = \Omega_\varepsilon$.

Algorithm 3 Perturbed LNS

Inputs: Logits $\boldsymbol{\theta} \in \mathbb{R}^d$, finite feasible set $\mathcal{Y} \subset \mathbb{R}^d$, noise distribution $\mathcal{D}$, temperature parameter $\varepsilon > 0$, initial solution $\boldsymbol{y}^{(0)} \in \mathcal{Y}$, number of iterations $K \in \mathbb{N}$.
for $k = 0$ to $K-1$ **do**
  Choose a subset of coordinates $S^{(k)} \subset [d]$.
  Define the local neighborhood: $\mathcal{Y}^{(k)} \triangleq \left\{\boldsymbol{y} \in \mathcal{Y} \mid \boldsymbol{y}_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]}\right\}$
  Sample $Z^{(k)} \sim \mathcal{D}$
  Set $\boldsymbol{y}^{(k+1)} \leftarrow \arg\max_{\boldsymbol{y}\in\mathcal{Y}^{(k)}}\langle\boldsymbol{\theta} + \varepsilon Z^{(k)}, \boldsymbol{y}\rangle$
end for
Return $\frac{1}{K}\sum_{k=1}^{K}\boldsymbol{y}^{(k)}$

4 Learning with Regularized Large Neighborhood Search

In this section, we show how to train models using RLNS as an output combinatorial layer.

4.1 RLNS-∞ and global Fenchel-Young loss

In this section, we consider the asymptotic regime where the number of RLNS iterations grows large ($K \to \infty$). Let $\Omega : \triangle^{\mathcal{Y}} \to \mathbb{R}$ be a global distribution-space regularizer, and $L_\Omega : \mathbb{R}^d \times \triangle^{\mathcal{Y}} \to \mathbb{R}_+$ be the corresponding global Fenchel-Young loss, as defined in Definition 1. For any target distribution[1] $\boldsymbol{p} \in \triangle^{\mathcal{Y}}$, the gradient of this loss is given by $\nabla_{\boldsymbol{\theta}} L_\Omega(\boldsymbol{\theta}; \boldsymbol{p}) = \mathbb{E}_{\boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})}[Y] - \mathbb{E}_{\boldsymbol{p}}[Y]$. Since the expectation of the global regularized distribution $\boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})$ is intractable, we can use the RLNS iterates $(\boldsymbol{y}^{(k)})_{k\in\mathbb{N}}$ of Algorithm 2 to construct an MCMC estimator of this gradient:

$$\nabla_{\boldsymbol{\theta}} L_\Omega(\boldsymbol{\theta}; \boldsymbol{p}) \approx \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{y}^{(k)} - \mathbb{E}_{\boldsymbol{p}}[Y].$$

The behavior and bias of this stochastic gradient estimator depend on the chosen regularization, variable selection scheme, and number of iterations K.

Shannon-regularized case. When $\Omega = -\gamma H^s$, Proposition 1 guarantees that the stationary distribution of the RLNS Markov chain exactly matches the target distribution, i.e., $\pi^{\mathrm{LNS}}_{\boldsymbol{\theta},\Omega} = \boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})$. In this setting, the only source of bias in the gradient estimator stems from the finite mixing time of the Markov chain. While biased for any fixed K, the estimator is *consistent* in the $K \to \infty$ limit.

[1] In practice, $\boldsymbol{p}$ is typically a Dirac distribution $\delta^{\mathcal{Y}}_{\boldsymbol{y}}$ on some target $\boldsymbol{y} \in \mathcal{Y}$, such that $\mathbb{E}_{\boldsymbol{p}}[Y] = \boldsymbol{y}$.

General case. For a general Ω, RLNS accumulates two sources of bias: one due to the mixing time of the chain, and one due to the intrinsic discrepancy between $\pi^{\mathrm{LNS}}_{\boldsymbol{\theta},\Omega}$ and $\boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})$. In Appendix A, we characterize the latter through the *local consistency gap*: the worst-case mismatch between a single RLNS update and the one of an ideal block Gibbs sampler targeting $\boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})$ (Definition 6). The bias scales with this gap, which in turn decreases as Ω approaches $-\gamma H^s$ (Propositions 8 and 9).

In the case of perturbed LNS, analyzing the problem's *constraint hypergraph* (Definition 7) shows that this gap vanishes when the feasible set decomposes into independent variable blocks aligned with the neighborhood selection, and otherwise grows with the number of constraints coupling frozen and free variables (Propositions 11 and 12). This motivates "min-cut" variable selection schemes that minimize shared constraints between frozen and free variables (Appendix A.4 and Fig. 3).

The choice of K. The number of iterations K allows us to interpolate between two learning paradigms. Setting $K = 1$ minimizes computational overhead and provides unbiased gradients for a local Fenchel-Young loss, a generalized form of pseudolikelihood (Section 4.2). Conversely, a larger K better estimates global Fenchel-Young gradients but incurs higher computational costs.

4.2 RLNS-1 and local Fenchel-Young loss

We study the special case of a single RLNS iteration. For any target solution $\boldsymbol{y} \in \mathcal{Y}$ and fixed coordinates $S \subseteq [d]$, we denote the corresponding neighborhood by $\mathcal{Y}_S \triangleq \{\boldsymbol{y}' \in \mathcal{Y} \mid \boldsymbol{y}'_{[S]} = \boldsymbol{y}_{[S]}\}$.

Definition 5 (Local Fenchel-Young loss)**.** *For any distribution $\nu \in \triangle^{\mathcal{P}([d])}$ on coordinate subsets, define the expected smoothed local maximum value function $F_{\boldsymbol{y}} : \mathbb{R}^{|\mathcal{Y}|} \to \mathbb{R} \cup \{+\infty\}$ as:*

$$F_{\boldsymbol{y}}(\boldsymbol{s}) \triangleq \mathbb{E}_{S\sim\nu}\left[f_{\Omega^{\mathcal{Y}_S}}\left(\boldsymbol{s}_{[\mathcal{Y}_S]}\right)\right].$$

Letting its convex conjugate be $\Omega_{\boldsymbol{y}} \triangleq (F_{\boldsymbol{y}})^*$, the corresponding target-dependent, local Fenchel-Young loss *$L_{\Omega_{\boldsymbol{y}}} : \mathbb{R}^d \times \mathrm{dom}(\Omega_{\boldsymbol{y}}) \to \mathbb{R}_+$ is given by:*

$$L_{\Omega_{\boldsymbol{y}}}(\boldsymbol{\theta}\,;\boldsymbol{p}) \triangleq F_{\boldsymbol{y}}(\boldsymbol{s}_{\boldsymbol{\theta}}) + \Omega_{\boldsymbol{y}}(\boldsymbol{p}) - \langle \boldsymbol{\theta}, \mathbb{E}_{\boldsymbol{p}}[Y]\rangle.$$

The following proposition, which we prove in Appendix B.7, shows we can obtain unbiased stochastic gradients for this loss directly from a single step of RLNS.

Proposition 4 (Unbiased gradients via RLNS-1)**.** *For any $\boldsymbol{y} \in \mathcal{Y}$, we have $\delta^{\mathcal{Y}}_{\boldsymbol{y}} \in \mathrm{dom}(\Omega_{\boldsymbol{y}})$. Moreover, the gradient of the local Fenchel-Young loss with target $\boldsymbol{y}$ is given by:*

$$\nabla_{\boldsymbol{\theta}} L_{\Omega_{\boldsymbol{y}}}(\boldsymbol{\theta}\,;\delta^{\mathcal{Y}}_{\boldsymbol{y}}) = \mathbb{E}_{S\sim\nu}\left[\mathbb{E}_{\boldsymbol{p}_{\Omega^{\mathcal{Y}_S}}\left((\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}_S]}\right)}[Y]\right] - \boldsymbol{y}.$$

Consequently, executing Algorithm 2 with ground-truth initialization $\boldsymbol{y}^{(0)} = \boldsymbol{y}$ and a single iteration $K = 1$ yields an unbiased estimator of $\nabla_{\boldsymbol{\theta}} L_{\Omega_{\boldsymbol{y}}}(\boldsymbol{\theta}\,;\delta^{\mathcal{Y}}_{\boldsymbol{y}})$.

Connection with pseudolikelihood. In the Shannon-regularized setting $\Omega = -\gamma H^s$, this local Fenchel-Young loss recovers the pseudolikelihood (a.k.a. composite likelihood) objective [Besag, 1974], which maximizes the likelihood of subsets of observed variables conditioned on the rest.

Proposition 5 (Equivalence to pseudolikelihood)**.** *When $\Omega = -\gamma H^s$, the local Fenchel-Young loss with target $\boldsymbol{y}$ equals the expected negative log-likelihood of $\boldsymbol{y}$ conditioned on $\boldsymbol{y}_{[S]}$:*

$$L_{\Omega_{\boldsymbol{y}}}(\boldsymbol{\theta}\,;\delta^{\mathcal{Y}}_{\boldsymbol{y}}) = -\gamma \cdot \mathbb{E}_{S\sim\nu}\left[\log\left(\boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})(\boldsymbol{y} \mid \boldsymbol{y}_{[S]})\right)\right].$$

The derivation is given in Appendix B.8. Thus, since the global Fenchel-Young loss is given by $L_\Omega(\boldsymbol{\theta}\,;\delta^{\mathcal{Y}}_{\boldsymbol{y}}) = -\gamma \log \boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})(\boldsymbol{y})$ in this setting, the number of entropic RLNS iterations enables interpolating between pseudolikelihood and exact maximum likelihood estimation.

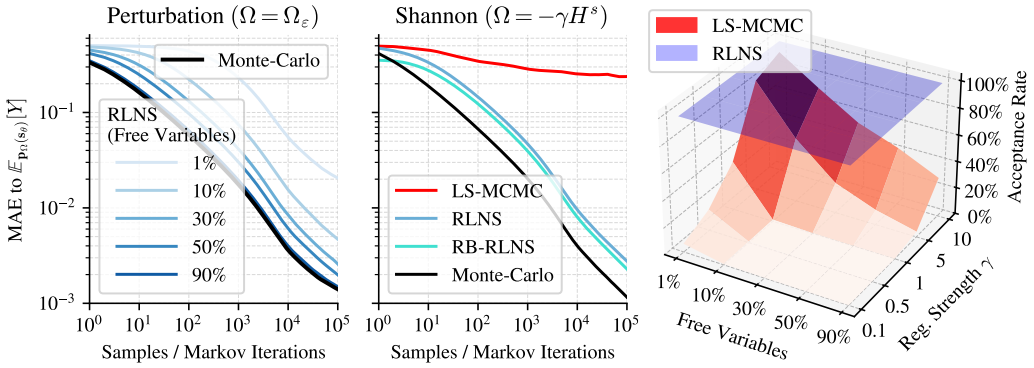


Figure 1: Estimation on κ-hot vectors. **Left:** convergence of RLNS with varying proportion of free variables compared to i.i.d. Monte-Carlo estimation. **Center:** convergence with 30% of free variables, against Local Search MCMC and i.i.d. Monte-Carlo estimation (RB- denotes Rao-Blackwellization). **Right:** acceptance rate with varying proportion of free variables and regularization strength γ.

5 Experiments

5.1 Estimation on κ-hot vectors

To validate our proposed estimators in a controlled setting, we consider the set of κ-hot vectors, $\mathcal{Y} \triangleq \{\boldsymbol{y} \in \{0,1\}^d \mid \sum_{i=1}^d y_i = \kappa\}$. In this context, the expectation of the global regularized distribution $\mathbb{E}_{\boldsymbol{p}_\Omega(\boldsymbol{s}_\theta)}[Y]$ can be exactly computed via regularized dynamic programming for entropic regularization [Vivier-Ardisson et al., 2026b], or accurately estimated via exact i.i.d. Monte-Carlo sampling for perturbation-based regularization. This provides a reliable ground truth against which we can evaluate our approach. Full experimental details are provided in Appendix C.

Baselines. We compare RLNS against exact i.i.d. Monte-Carlo estimation (sampling directly from the global perturb-and-MAP [Papandreou and Yuille, 2011, Berthet et al., 2020] or Gibbs distribution) and Local Search MCMC [Vivier-Ardisson et al., 2026a] using the same neighborhoods. Drawing exact global Monte-Carlo samples requires a global oracle, which is tractable here on κ-hot vectors by design, but intractable on general NP-hard problems. Thus, i.i.d. Monte-Carlo serves as an idealized upper bound on sample efficiency.

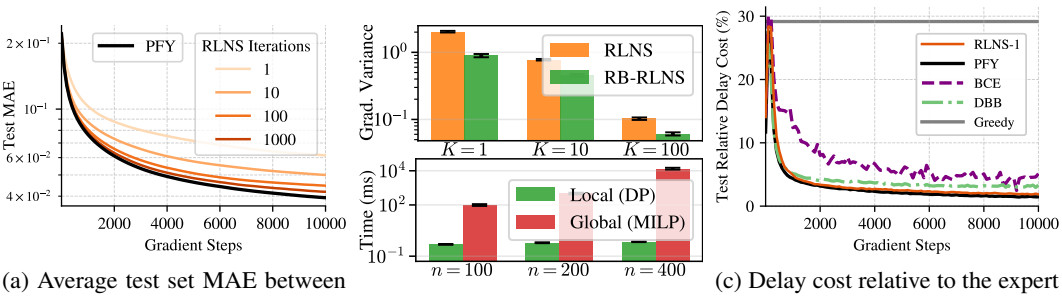
Results. Tracking the mean absolute error (MAE) to $\mathbb{E}_{\boldsymbol{p}_\Omega(\boldsymbol{s}_\theta)}[Y]$ (Fig. 1), i.i.d. Monte-Carlo naturally converges faster per sample than RLNS, whose Markov chain iterates are correlated and update only free variables via local oracles. Leaving more variables free accelerates RLNS convergence toward this global ceiling at the cost of larger subproblems (Fig. 1, left). Moreover, RLNS converges much faster than Local Search MCMC (Fig. 1, center), as its 100% acceptance rate avoids Metropolis-Hastings stagnation on larger neighborhoods or low γ (Fig. 1, right), and Rao-Blackwellization further reduces early-iteration error (Fig. 1, center).

5.2 Generalized assignment problem

We now evaluate RLNS as a combinatorial layer on the generalized assignment problem (GAP) [Martello and Toth, 1992]. In the GAP, we are given n items and m bins with capacity $C_j \in \mathbb{N}$. Each item-bin assignment has value $\theta_{ij} \in \mathbb{R}$ and a weight $w_{ij} \in \mathbb{N}$. The combinatorial problem writes:

$$\max_{\boldsymbol{y} \in \{0,1\}^{n \times m}} \sum_{i=1}^{n} \sum_{j=1}^{m} \theta_{ij} y_{ij} \quad \text{s.t.} \quad \sum_{j=1}^{m} y_{ij} \leq 1 \quad \forall i \in [n], \quad \text{and} \quad \sum_{i=1}^{n} w_{ij} y_{ij} \leq C_j \quad \forall j \in [m].$$

The first constraint ensures each item is assigned to at most one bin, while the second one ensures that the capacity of each bin is not exceeded. Given a dataset of $(\boldsymbol{x}, \boldsymbol{y})$ pairs where $\boldsymbol{x} \in \mathbb{R}^{n \times m \times p}$ contains relational features for each item-bin pair, the goal is to learn a neural network $f_W : \mathbb{R}^p \to \mathbb{R}$ that predicts assignment scores $\theta_{ij} = f_W(\boldsymbol{x}_{ij})$, such that $\hat{\boldsymbol{y}}(\boldsymbol{\theta}) \approx \boldsymbol{y}$.



(a) Average test set MAE between predicted and ground-truth solutions of RLNS and PFY on the GAP.

(b) Variance of RLNS gradients and oracle wall-clock time on the GAP.

(c) Delay cost relative to the expert policy on stochastic vehicle scheduling test set instances.

Figure 2: Empirical evaluation of RLNS as a combinatorial layer on the generalized assignment problem (a, b) and stochastic vehicle scheduling (c).

Local oracles. We choose to freeze the assignments to all bins except a randomly selected one at each RLNS iteration. The LNS subproblem is then a $0/1$ one-dimensional Knapsack problem, which admits an exact dynamic programming solution [Kellerer et al., 2004], and we can sample from the local regularized distributions via perturbed or regularized dynamic programming [Vivier-Ardisson et al., 2026b]. Detailed explanations and full experimental details are given in Appendix D.

Results. In Fig. 2a, we intentionally compare perturbed RLNS directly against global perturbed Fenchel-Young learning (PFY, Berthet et al. [2020]) to isolate how the number of local updates bridges local and global learning: increasing K steadily lowers test MAE and closes the gap to global PFY despite the stationary discrepancy of perturbed LNS. Increasing K also reduces stochastic gradient variance, which is further reduced under entropic regularization via Rao-Blackwellization (Fig. 2b, top). Finally, comparing the wall-clock time of a single oracle call across problem scales ($m = n/10$), local DP-based MAP oracles are orders of magnitude faster than global MILP solvers, with the speedup growing sharply with n (Fig. 2b, bottom).

5.3 Stochastic vehicle scheduling

We now evaluate RLNS as a combinatorial layer on a stochastic vehicle scheduling benchmark [Parmentier, 2022, Dalle et al., 2022, Hoppe et al., 2025]. The objective is to route a fleet of vehicles in order to serve a set of spatio-temporal tasks, minimizing the sum of the deterministic vehicle cost and the expected delay cost across a set of stochastic scenarios. The problem is framed as finding paths on a directed acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where a feasible routing $\boldsymbol{y} \in \{0,1\}^{|\mathcal{E}|}$ satisfies flow conservation and unit demand constraints. To imitate the ground-truth expert (the optimal solution of a two-stage stochastic MILP with full scenario knowledge), a feed-forward neural network predicts edge costs θ_e from edge-level features, and the predicted routing is obtained via a flow LP. Complete experimental details are provided in Appendix E.

Approach and baselines. We use RLNS-1 with perturbation-based regularization. At each step, we freeze the edges corresponding to 66% of randomly selected vehicle routes, and exactly re-optimize the remaining free variables using a significantly smaller flow LP. We compare RLNS against global perturbed Fenchel-Young learning (PFY, Berthet et al. [2020]), blackbox differentiation (DBB, Vlastelica et al. [2020]), an unstructured binary cross-entropy loss (BCE, see, e.g., Joshi et al. [2019]), and a non-learned greedy policy that minimizes only the deterministic vehicle cost.

Results. All learned methods quickly reach the optimal vehicle cost by using the same number of vehicles as the expert. However, in terms of delay cost (Fig. 2c), RLNS-1 outperforms DBB and BCE and nearly matches global PFY while only solving flow subproblems of roughly one-third the original size. On these $n = 50$ instances (the largest scale tractable for the ground-truth two-stage stochastic MILP used to generate data), a single local MAP call averages 1.8 ms versus 4.4 ms for the global MAP oracle, a gap that widens as problem size grows.

Conclusion

We introduced regularized large neighborhood search (RLNS), turning standard LNS heuristics into rejection-free MCMC samplers to provide gradients through combinatorial optimization layers using only scalable local oracles, and interpolating between local composite likelihood ($K = 1$) and global Fenchel-Young learning ($K \to \infty$).

Bridging communities and future work. While we exhibit links between LNS and Gibbs sampling and prior work bridged local search algorithms with Metropolis-Hastings [Vivier-Ardisson et al., 2026a], we believe more combinatorial optimization metaheuristics can be revisited from a sampling perspective. For instance, LNS can use local search heuristics as approximate subproblem solvers in practice [Pisinger and Ropke, 2018], suggesting a natural bridge between "local-search-within-LNS" metaheuristics and "Metropolis-within-Gibbs" samplers [Roberts and Rosenthal, 1997, Ascolani et al., 2026]. Differently, the systematic design of optimal variable selection schemes tailored to a given problem structure and regularization choice remains an important open direction.

Limitations. First, RLNS faces an oracle / exactness trade-off: entropic RLNS guarantees exact block Gibbs sampling equivalence but requires local marginal oracles, whereas perturbed RLNS relies only on local MAP oracles at the cost of a stationary bias when constraints couple frozen and free variables (Appendix A), for which our theoretical bounds are structural rather than numerically tight. Second, our learning formulation focuses on Fenchel-Young losses, and the unbiased $K = 1$ regime requires target solutions $\boldsymbol{y} \in \mathcal{Y}$ for initialization, which may not be directly available in contexts such as reinforcement or decision-focused learning without supervision.

Acknowledgments and disclosures

This work was fully supported by Google DeepMind, where Germain Vivier-Ardisson is a full-time employee and PhD candidate. The authors have no competing interests or financial relationships outside the submitted work to disclose.

References

A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Z. Kolter. Differentiable Convex Optimization Layers. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/9ce3c52fc54362e22053399d3181c638-Abstract.html.

C. Archetti and M. G. Speranza. A survey on matheuristics for routing problems. *EURO Journal on Computational Optimization*, 2(4):223–246, 2014.

F. Ascolani, G. O. Roberts, and G. Zanella. Scalability of Metropolis-within-Gibbs schemes for high-dimensional Bayesian models, Jan. 2026. URL http://arxiv.org/abs/2403.09416. arXiv:2403.09416 [stat].

A. Asuncion, Q. Liu, A. Ihler, and P. Smyth. Learning with Blocks: Composite Likelihood and Contrastive Divergence. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 33–40. JMLR Workshop and Conference Proceedings, Mar. 2010. URL https://proceedings.mlr.press/v9/asuncion10a.html.

G. Bertasius, Q. Liu, L. Torresani, and J. Shi. Local Perturb-and-MAP for Structured Prediction. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 585–594. PMLR, Apr. 2017. URL https://proceedings.mlr.press/v54/bertasius17a.html.

Q. Berthet, M. Blondel, O. Teboul, M. Cuturi, J.-P. Vert, and F. Bach. Learning with Differentiable Perturbed Optimizers. In *Advances in Neural Information Processing Systems*, volume 33, pages 9508–9519. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/6bb56208f672af0dd65451f869fedfd9-Abstract.html.

J. Besag. Spatial Interaction and the Statistical Analysis of Lattice Systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):192–236, 1974. ISSN 0035-9246. URL https://www.jstor.org/stable/2984812.

J. Besag. On the Statistical Analysis of Dirty Pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, 48(3):259–302, 1986. ISSN 0035-9246. URL https://www.jstor.org/stable/2345426.

M. Blondel and V. Roulet. The Elements of Differentiable Programming, June 2025. URL http://arxiv.org/abs/2403.14606. arXiv:2403.14606 [cs].

M. Blondel, A. F. T. Martins, and V. Niculae. Learning with Fenchel-Young losses. *Journal of Machine Learning Research*, 21(35):1–69, 2020. ISSN 1533-7928. URL http://jmlr.org/papers/v21/19-021.html.

M. Blondel, Q. Berthet, M. Cuturi, R. Frostig, S. Hoyer, F. Llinares-Lopez, F. Pedregosa, and J.-P. Vert. Efficient and Modular Implicit Differentiation. In *Advances in Neural Information Processing Systems*, volume 35, pages 5230–5242. Curran Associates, Inc., 2022. doi: 10.52202/068431-0378. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/228b9279ecf9bbafe582406850c57115-Abstract-Conference.html.

L. Bouvier, T. Prunet, V. Leclère, and A. Parmentier. Primal-dual algorithm for contextual stochastic combinatorial optimization, May 2025. URL http://arxiv.org/abs/2505.04757. arXiv:2505.04757 [cs].

G. Dalle, L. Baty, L. Bouvier, and A. Parmentier. Learning with Combinatorial Optimization Layers: a Probabilistic Approach, Dec. 2022. URL http://arxiv.org/abs/2207.13513. arXiv:2207.13513 [cs, math, stat].

J. M. Danskin. The Theory of Max-Min, with Applications. *SIAM Journal on Applied Mathematics*, 14(4):641–664, July 1966. ISSN 0036-1399. doi: 10.1137/0114053. URL https://epubs.siam.org/doi/abs/10.1137/0114053.

A. Ferber, B. Wilder, B. Dilkina, and M. Tambe. MIPaaL: Mixed Integer Program as a Layer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(02):1504–1511, Apr. 2020. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v34i02.5509. URL https://ojs.aaai.org/index.php/AAAI/article/view/5509.

E. Gselmann. Entropy functions and functional equations. *Mathematical Communications*, 16(2): 347–357, 2011.

G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

H. Hoppe, L. Baty, L. Bouvier, A. Parmentier, and M. Schiffer. Structured Reinforcement Learning for Combinatorial Decision-Making. In *Advances in Neural Information Processing Systems*, volume 38, pages 173726–173761. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/fe100e5e7bca984803f0e4c6e94a6a88-Abstract-Conference.html.

Y. Horibe. Entropy of terminal distributions and the Fibonacci trees. *The Fibonacci Quarterly*, 26(2): 135–140, 1988.

A. Hyvärinen. Consistency of Pseudolikelihood Estimation of Fully Visible Boltzmann Machines. *Neural Computation*, 18(10):2283–2292, Oct. 2006. ISSN 0899-7667. doi: 10.1162/neco.2006.18.10.2283. URL https://doi.org/10.1162/neco.2006.18.10.2283.

C. K. Joshi, T. Laurent, and X. Bresson. An Efficient Graph Convolutional Network Technique for the Travelling Salesman Problem, Oct. 2019. URL http://arxiv.org/abs/1906.01227. arXiv:1906.01227 [cs].

H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer, Berlin, Heidelberg, 2004. ISBN 978-3-642-07311-3 978-3-540-24777-7. doi: 10.1007/978-3-540-24777-7. URL http://link.springer.com/10.1007/978-3-540-24777-7.

R. Krishnan, S. Lacoste-Julien, and D. Sontag. Barrier Frank-Wolfe for Marginal Inference. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/hash/0c74b7f78409a4022a2c4c5a5ca3ee19-Abstract.html.

J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558607781.

D. Levin, Y. Peres, and E. Wilmer. *Markov Chains and Mixing Times*. American Mathematical Society, Providence, Rhode Island, Dec. 2008. ISBN 978-0-8218-4739-8 978-1-4704-1204-3. doi: 10.1090/mbk/058.

Z. Li and J. Eisner. First- and second-order expectation semirings with applications to minimum-risk training on translation forests. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 40–51, USA, Aug. 2009. Association for Computational Linguistics. ISBN 978-1-932432-59-6. URL https://dl.acm.org/doi/10.5555/1699510.1699517.

B. G. Lindsay. Composite likelihood methods. In *Statistical Inference from Stochastic Processes: Proceedings of the AMS-IMS-SIAM Joint Summer Research Conference Held August 9-15, 1987, with Support from the National Science Foundation and the Army Research Office*, volume 80, page 221. American Mathematical Soc., 1988.

J. Mandi and T. Guns. Interior Point Solving for LP-based prediction+optimisation. In *Advances in Neural Information Processing Systems*, volume 33, pages 7272–7282. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/51311013e51adebc3c34d2cc591fefee-Abstract.html.

J. Mandi, J. Kotary, S. Berden, M. Mulamba, V. Bucarey, T. Guns, and F. Fioretto. Decision-Focused Learning: Foundations, State of the Art, Benchmark and Future Opportunities. *Journal of Artificial Intelligence Research*, 80:1623–1701, Aug. 2024. ISSN 1076-9757. doi: 10.1613/jair.1.15320. URL http://arxiv.org/abs/2307.13565. arXiv:2307.13565 [cs, math].

S. Martello and P. Toth. Generalized assignment problems. In T. Ibaraki, Y. Inagaki, K. Iwama, T. Nishizeki, and M. Yamashita, editors, *Algorithms and Computation*, pages 351–369, Berlin, Heidelberg, 1992. Springer. ISBN 978-3-540-47501-9. doi: 10.1007/3-540-56279-6_88.

A. Martins and R. Astudillo. From Softmax to Sparsemax: A Sparse Model of Attention and Multi-Label Classification. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1614–1623. PMLR, June 2016. URL https://proceedings.mlr.press/v48/martins16.html.

I. W. McKeague and W. Wefelmeyer. Markov chain Monte Carlo and Rao–Blackwellization. *Journal of Statistical Planning and Inference*, 85(1):171–182, Apr. 2000. ISSN 0378-3758. doi: 10.1016/S0378-3758(99)00079-8. URL https://www.sciencedirect.com/science/article/pii/S0378375899000798.

D. McKenzie, S. W. Fung, and H. Heaton. Differentiating Through Integer Linear Programs with Quadratic Regularization and Davis-Yin Splitting. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=KxBQPz7HKh.

A. Mensch and M. Blondel. Differentiable Dynamic Programming for Structured Prediction and Attention. In *Proceedings of the 35th International Conference on Machine Learning*, pages 3462–3471. PMLR, July 2018. URL https://proceedings.mlr.press/v80/mensch18a.html.

A. Y. Mitrophanov. Sensitivity and Convergence of Uniformly Ergodic Markov Chains. *Journal of Applied Probability*, 42(4):1003–1014, 2005. ISSN 0021-9002. URL https://www.jstor.org/stable/30040824. Publisher: Applied Probability Trust.

V. Niculae, A. Martins, M. Blondel, and C. Cardie. SparseMAP: Differentiable Sparse Structured Inference. In *Proceedings of the 35th International Conference on Machine Learning*, pages 3799–3808. PMLR, July 2018. URL https://proceedings.mlr.press/v80/niculae18a.html.

G. Papandreou and A. L. Yuille. Perturb-and-MAP random fields: Using discrete optimization to learn and sample from energy models. In *2011 International Conference on Computer Vision*, pages 193–200, Nov. 2011. doi: 10.1109/ICCV.2011.6126242. URL https://ieeexplore.ieee.org/document/6126242. ISSN: 2380-7504.

A. Parmentier. Learning to Approximate Industrial Problems by Operations Research Classic Problems. *Operations Research*, 70(1):606–623, Jan. 2022. ISSN 0030-364X. doi: 10.1287/opre.2020.2094. URL https://pubsonline.informs.org/doi/10.1287/opre.2020.2094.

D. Pisinger and S. Ropke. Large neighborhood search. In *Handbook of Metaheuristics*, pages 99–127. Springer, 2018.

G. Roberts and J. Rosenthal. Geometric Ergodicity and Hybrid Markov Chains. *Electronic Communications in Probability*, 2(none):13–25, Jan. 1997. ISSN 1083-589X, 1083-589X. doi: 10.1214/ECP.v2-981. URL https://projecteuclid.org/journals/electronic-communications-in-probability/volume-2/issue-none/Geometric-Ergodicity-and-Hybrid-Markov-Chains/10.1214/ECP.v2-981.full.

R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970. ISBN 978-0-691-01586-6. URL https://www.jstor.org/stable/j.ctt14bs1ff.

U. Sadana, A. Chenreddy, E. Delage, A. Forel, E. Frejinger, and T. Vidal. A survey of contextual optimization methods for decision-making under uncertainty. *European Journal of Operational Research*, 320(2):271–289, Jan. 2025. ISSN 0377-2217. doi: 10.1016/j.ejor.2024.03.020. URL https://www.sciencedirect.com/science/article/pii/S0377221724002200.

M. Schiffer, H. Hoppe, Y. Su, L. Bouvier, and A. Parmentier. Combinatorial Optimization Augmented Machine Learning, Jan. 2026. URL http://arxiv.org/abs/2601.10583. arXiv:2601.10583 [cs].

C. Varin, N. Reid, and D. Firth. An Overview of Composite Likelihood Methods. *Statistica Sinica*, 21(1):5–42, 2011. ISSN 1017-0405. URL https://www.jstor.org/stable/24309261.

G. Vivier-Ardisson, M. Blondel, and A. Parmentier. Learning with Local Search MCMC Layers. *Transactions on Machine Learning Research*, 2026a. ISSN 2835-8856. URL https://openreview.net/forum?id=M2aOmbACNf.

G. Vivier-Ardisson, M. E. Sander, A. Parmentier, and M. Blondel. Differentiable Knapsack and Top-k Operators via Dynamic Programming. In *Advances in Neural Information Processing Systems*, volume 39. Curran Associates, Inc., 2026b. URL https://arxiv.org/abs/2601.21775.

M. Vlastelica, A. Paulus, V. Musil, G. Martius, and M. Rolínek. Differentiation of Blackbox Combinatorial Solvers. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BkevoJSYPB.

M. J. Wainwright and M. I. Jordan. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, Nov. 2008. ISSN 1935-8237, 1935-8245. doi: 10.1561/2200000001. URL https://www.nowpublishers.com/article/Details/MAL-001.

Notation

- We denote $\mathbb{R}_+ \triangleq [0, +\infty)$, $[D] \triangleq \{1, \dots, D\}$ for any integer $D \in \mathbb{N}$, and $\mathcal{P}(S)$ for the power set of any set S. We denote by $\mathbf{0}$, $\mathbf{1}$, and $\boldsymbol{I}$ the all-zeros vector, all-ones vector, and identity matrix (of appropriate dimensions).
- For any finite index set $\mathcal{Z}$ (such as $[D]$), vector $\boldsymbol{x} = (x_z)_{z \in \mathcal{Z}} \in \mathbb{R}^{|\mathcal{Z}|}$, and subset $I \subseteq \mathcal{Z}$, we denote $\boldsymbol{x}_{[I]} \triangleq (x_i)_{i \in I} \in \mathbb{R}^{|I|}$. Similarly, for any matrix $\boldsymbol{A} \in \mathbb{R}^{M \times D}$ and subsets $I \subseteq [M]$, $J \subseteq [D]$, we denote $\boldsymbol{A}_{[I,J]} \triangleq (A_{i,j})_{i \in I, j \in J} \in \mathbb{R}^{|I| \times |J|}$ (and $\boldsymbol{A}_{[:\,,j]} \triangleq \boldsymbol{A}_{[[M],\{j\}]}$).
- For any finite set $\mathcal{Y}$, we denote by $\triangle^{\mathcal{Y}} \triangleq \{\boldsymbol{p} \in \mathbb{R}_+^{|\mathcal{Y}|} \mid \sum_{\boldsymbol{y} \in \mathcal{Y}} p_{\boldsymbol{y}} = 1\}$ the set of categorical distributions on $\mathcal{Y}$. For $\boldsymbol{p} \in \triangle^{\mathcal{Y}}$, we equivalently write $\boldsymbol{p}(\boldsymbol{y})$ or $p_{\boldsymbol{y}}$ for the probability of $\boldsymbol{y} \in \mathcal{Y}$, $\operatorname{supp}(\boldsymbol{p}) \triangleq \{\boldsymbol{y} \in \mathcal{Y} \mid p_{\boldsymbol{y}} > 0\}$ for its support, and $\delta_{\boldsymbol{y}}^{\mathcal{Y}} \in \triangle^{\mathcal{Y}}$ for the Dirac distribution at $\boldsymbol{y} \in \mathcal{Y}$.
- For any subset of coordinates $I \subseteq [d]$, any solution $\boldsymbol{y}' \in \mathcal{Y}$ and $\boldsymbol{y} \in \mathcal{Y}$ such that $\boldsymbol{y}_{[I]} = \boldsymbol{y}'_{[I]}$, we denote by:

$$\boldsymbol{p}(\boldsymbol{y} \mid \boldsymbol{y}'_{[I]}) \triangleq \frac{\boldsymbol{p}(\boldsymbol{y})}{\sum\limits_{\substack{\boldsymbol{y}'' \in \mathcal{Y} \\ \text{s.t. } \boldsymbol{y}''_{[I]} = \boldsymbol{y}'_{[I]}}} \boldsymbol{p}(\boldsymbol{y}'')}$$

  the corresponding conditional probability (we always use the functional notation for conditional probabilities).
- For any $\boldsymbol{p}, \boldsymbol{q} \in \triangle^{\mathcal{Y}}$, we denote their total variation (TV) distance, and for $\boldsymbol{q} \in \operatorname{ri}(\triangle^{\mathcal{Y}})$ their Kullback-Leibler (KL) divergence (with the convention $0 \log 0 = 0$), by:

$$d_{\mathrm{TV}}(\boldsymbol{p}, \boldsymbol{q}) \triangleq \frac{1}{2} \sum_{\boldsymbol{y} \in \mathcal{Y}} |p_{\boldsymbol{y}} - q_{\boldsymbol{y}}|, \qquad D_{\mathrm{KL}}(\boldsymbol{p} \| \boldsymbol{q}) \triangleq \sum_{\boldsymbol{y} \in \mathcal{Y}} p_{\boldsymbol{y}} \log \left(\frac{p_{\boldsymbol{y}}}{q_{\boldsymbol{y}}} \right).$$

- We denote the complement of any event E by $\overline{E}$, and its indicator by $\mathbb{1}_E \in \{0, 1\}$.
- For any function $f : \mathbb{R}^D \to \mathbb{R} \cup \{+\infty\}$, we denote by $\operatorname{dom}(f) \triangleq \{\boldsymbol{x} \in \mathbb{R}^D \mid f(\boldsymbol{x}) < +\infty\}$ its domain, and its convex conjugate by

$$f^*(\boldsymbol{y}) \triangleq \sup_{\boldsymbol{x} \in \operatorname{dom}(f)} \{\langle \boldsymbol{x}, \boldsymbol{y} \rangle - f(\boldsymbol{x})\}.$$

- For any convex function $f : \mathcal{C} \to \mathbb{R}$ on a convex set $\mathcal{C} \subseteq \mathbb{R}^D$ that is differentiable on $\operatorname{ri}(\mathcal{C})$, we denote the Bregman divergence generated by f as

$$B_f(\boldsymbol{x} \| \boldsymbol{y}) \triangleq f(\boldsymbol{x}) - f(\boldsymbol{y}) - \langle \nabla f(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle \quad \forall (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{C} \times \operatorname{ri}(\mathcal{C}).$$

- For any function $F : \mathbb{R}^D \to \mathbb{R}^N$, we denote by $\operatorname{range}(F) \triangleq \{F(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathbb{R}^D\}$ its range.
- We denote the convex hull of any set $\mathcal{Y} \subseteq \mathbb{R}^D$ by $\operatorname{conv}(\mathcal{Y})$, and the relative interior of a convex set $\mathcal{C} \subseteq \mathbb{R}^D$ by $\operatorname{ri}(\mathcal{C})$.

A Geometrical insights and approximation bounds for general regularizers

When $\Omega \neq -\gamma H^s$, the locally regularized distributions used by RLNS may differ from the conditional distributions of the global smoothed maximizer $\boldsymbol{p}_\Omega(\boldsymbol{s_\theta})$, thereby precluding the block Gibbs sampling perspective established in Proposition 1. In this section, we give geometrical insights into this more general case, and establish bounds on the distance between the algorithm's stationary distribution $\pi^{\mathrm{LNS}}_{\boldsymbol{\theta},\Omega}$ and the target $\boldsymbol{p}_\Omega(\boldsymbol{s_\theta})$.

A.1 Geometrical insights

First, we show that the update distributions of RLNS and the exact Gibbs sampler with target $\boldsymbol{p}_\Omega(\boldsymbol{s_\theta}) \in \triangle^{\mathcal{Y}}$ are in fact two different projections of the latter onto the local simplex $\triangle^{\mathcal{Y}^{(k)}}$.

Proposition 6 (RLNS and Gibbs updates as Bregman projections)**.** *Let $\Omega : \triangle^{\mathcal{Y}} \to \mathbb{R}$ be a l.s.c. Legendre-type regularizer, $\mathcal{Y}^{(k)} \subseteq \mathcal{Y}$ be the neighborhood used by Algorithm 2 at step k, and $B_\Omega : \triangle^{\mathcal{Y}} \times \mathrm{ri}(\triangle^{\mathcal{Y}}) \to \mathbb{R}_+$ denote the Bregman divergence generated by Ω. The local RLNS update distribution corresponds to the projection of the global smoothed maximizer $\boldsymbol{p}_\Omega(\boldsymbol{s_\theta})$ onto the local simplex $\triangle^{\mathcal{Y}^{(k)}}$ with respect to B_Ω:*

$$\boldsymbol{p}_{\Omega^{\mathcal{Y}^{(k)}}}\left((\boldsymbol{s_\theta})_{[\mathcal{Y}^{(k)}]}\right) = \operatorname*{arg\,min}_{\boldsymbol{q}\in\triangle^{\mathcal{Y}^{(k)}}} B_\Omega\left(\iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q}) \| \boldsymbol{p}_\Omega(\boldsymbol{s_\theta})\right).$$

Similarly, the update distribution of the ideal block Gibbs sampler with target $\boldsymbol{p}_\Omega(\boldsymbol{s_\theta})$ is its projection onto the local simplex with respect to the Kullback-Leibler divergence:

$$\boldsymbol{p}_\Omega(\boldsymbol{s_\theta})\left(\cdot \mid \boldsymbol{y}^{(k)}_{[S^{(k)}]}\right) = \operatorname*{arg\,min}_{\boldsymbol{q}\in\triangle^{\mathcal{Y}^{(k)}}} D_{\mathrm{KL}}\left(\iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q}) \| \boldsymbol{p}_\Omega(\boldsymbol{s_\theta})\right).$$

Proposition 6, proved in Appendix B.6, shows that RLNS and the ideal Gibbs sampler update distributions both correspond to Bregman projections of the same target distribution $\boldsymbol{p}_\Omega(\boldsymbol{s_\theta})$ onto the local simplex $\triangle^{\mathcal{Y}^{(k)}}$, simply with distinct geometries. While RLNS uses the same regularization Ω as the one used by the global target distribution, Gibbs sampling uses Shannon's negative entropy.

This gives a geometrical point of view on the claim of Proposition 1. Indeed, since we have $D_{\mathrm{KL}} = \frac{1}{\gamma} B_{-\gamma H^s}$ where $B_{-\gamma H^s}$ is the Bregman divergence generated by Shannon's scaled negative entropy, the corresponding projection problems are equivalent when $\Omega = -\gamma H^s$ in the first place: the B_Ω-projections match D_{KL}-projections, and RLNS matches an exact Gibbs sampler for $\boldsymbol{p}_\Omega(\boldsymbol{s_\theta})$.

We now give a block-coordinate ascent (BCA) perspective on the expected RLNS updates.

Proposition 7 (Expected RLNS as block-coordinate ascent)**.** *Assume that $\mathcal{Y} \subseteq \{0,1\}^d$, and let $\mathcal{M} \triangleq \mathrm{conv}(\mathcal{Y})$. Let $\Omega : \triangle^{\mathcal{Y}} \to \mathbb{R}$ be a l.s.c. strictly convex distribution-space regularizer. Its mean-space counterpart $\Omega_{\mathcal{M}} : \mathcal{M} \to \mathbb{R}$, defined as:*

$$\Omega_{\mathcal{M}}(\boldsymbol{\mu}) \triangleq \min_{\boldsymbol{p}\in\triangle^{\mathcal{Y}} \text{ s.t. } \mathbb{E}_{\boldsymbol{p}}[Y]=\boldsymbol{\mu}} \Omega(\boldsymbol{p}),$$

is strictly convex on $\mathcal{M}$. Furthermore, for any current solution $\boldsymbol{y}^{(k)} \in \mathcal{Y}$ and subset of frozen coordinates $S^{(k)} \subseteq [d]$, the expected RLNS local update matches the exact BCA update of the regularized mean-space objective over the free variables:

$$\mathbb{E}_{\boldsymbol{p}_{\Omega^{\mathcal{Y}^{(k)}}}\left((\boldsymbol{s_\theta})_{[\mathcal{Y}^{(k)}]}\right)}[Y] = \operatorname*{arg\,max}_{\boldsymbol{\mu}\in\mathcal{M} \text{ s.t. } \boldsymbol{\mu}_{[S^{(k)}]}=\boldsymbol{y}^{(k)}_{[S^{(k)}]}} \left\{\langle \boldsymbol{\theta}, \boldsymbol{\mu}\rangle - \Omega_{\mathcal{M}}(\boldsymbol{\mu})\right\}.$$

The proof is given in Appendix B.5.

A.2 Local consistency gap

We first define the *local consistency gap*, which measures the maximum discrepancy between an RLNS update and the ideal Gibbs sampling update that would target the global regularized distribution.

Definition 6 (Local consistency gap)**.** *Let $\boldsymbol{y}^{(k)} \in \mathcal{Y}$, $S^{(k)} \subseteq [d]$, and let $\mathcal{Y}^{(k)}$ denote the corresponding subset of feasible solutions, as defined in Algorithms 1 and 2.*

We define the local consistency gap *$\delta_{\boldsymbol{\theta},\Omega}$ as the maximum TV distance between the global regularized distribution $\boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})$ conditioned on the subset of structures matching $\boldsymbol{y}^{(k)}$ on coordinates $S^{(k)}$ and the local RLNS update distribution (seen as two elements of $\triangle^{\mathcal{Y}^{(k)}}$):*

$$\delta_{\boldsymbol{\theta},\Omega} \triangleq \max_{\boldsymbol{y}^{(k)},S^{(k)}} d_{\mathrm{TV}}\left(\boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})(\cdot \mid \boldsymbol{y}^{(k)}_{[S^{(k)}]}), \boldsymbol{p}_{\Omega^{\mathcal{Y}^{(k)}}}((\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]})\right).$$

As shown in Section 3.2, if $\Omega = -\gamma H^s$ for some $\gamma > 0$, the local updates match the conditionals exactly, implying $\delta_{\boldsymbol{\theta},\Omega} = 0$. For general regularizers, $\delta_{\boldsymbol{\theta},\Omega} \geq 0$. We now bound the total bias of the algorithm.

Proposition 8 (Bias of RLNS)**.** *Let P be the transition kernel of the RLNS Markov chain defined in Algorithm 2. Assuming $\mathrm{supp}(\boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})) = \mathcal{Y}$, let P^* be the transition kernel of the ideal Gibbs sampler targeting $\boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})$, i.e., which uses updates of the form:*

$$\boldsymbol{y}^{(k+1)} \sim \boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})\left(\cdot \mid \boldsymbol{y}^{(k)}_{[S^{(k)}]}\right).$$

Let $\tau_m(P^*)$ be the Dobrushin contraction coefficient of $(P^*)^m$, i.e.:

$$\tau_m(P^*) \triangleq \max_{\boldsymbol{y},\boldsymbol{y}' \in \mathcal{Y}} d_{\mathrm{TV}}\left((P^*)^m(\boldsymbol{y},\cdot), (P^*)^m(\boldsymbol{y}',\cdot)\right).$$

Let $m \geq 1$ such that $\tau_m(P^*) < 1$. Then, the distance between the RLNS stationary distribution $\pi^{\mathrm{LNS}}_{\boldsymbol{\theta},\Omega}$ and the target is bounded by:

$$d_{\mathrm{TV}}\left(\pi^{\mathrm{LNS}}_{\boldsymbol{\theta},\Omega}, \boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})\right) \leq \frac{m}{1-\tau_m(P^*)}\delta_{\boldsymbol{\theta},\Omega}.$$

Proposition 8, which is proved in Appendix B.9, enables us to now focus on bounding the local consistency gap $\delta_{\boldsymbol{\theta},\Omega}$ as it translates to a global bound on the distance between $\pi^{\mathrm{LNS}}_{\boldsymbol{\theta},\Omega}$ and $\boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})$.

We first bound the local consistency gap under the assumption that the deviation of the regularization from the scaled Shannon's negative entropy $\Omega - (-\gamma H^s)$ has a Lipschitz gradient with respect to the ℓ_1 norm.

Proposition 9 (Consistency gap bound via relative smoothness)**.** *Let $\gamma > 0$ be a reference temperature, and $h_\gamma : \triangle^{\mathcal{Y}} \to \mathbb{R}$ be the deviation of the regularization Ω from the scaled Shannon's negative entropy:*

$$h_\gamma(\boldsymbol{p}) \triangleq \Omega(\boldsymbol{p}) - (-\gamma H^s)(\boldsymbol{p}) = \Omega(\boldsymbol{p}) + \gamma \sum_{\boldsymbol{y} \in \mathcal{Y}} p_{\boldsymbol{y}} \log p_{\boldsymbol{y}}.$$

If h_γ is $L_{\Omega,\gamma}$-smooth with respect to the ℓ_1 norm, we have:

$$\delta_{\boldsymbol{\theta},\Omega} \leq \frac{L_{\Omega,\gamma}}{\gamma}.$$

The proof is given in Appendix B.10, and leverages the characterization of the RLNS and Gibbs updates as Bregman projections given in Proposition 6.

We now bound the consistency gap $\delta_{\boldsymbol{\theta},\Omega}$ in the special case of perturbation-based regularizers.

Proposition 10 (Consistency gap bound in the perturbation-based setting)**.** *Let Ω_ε be a perturbation-based regularization function, as defined in Definition 4. Let $\boldsymbol{p}_{\Omega_\varepsilon}(\boldsymbol{s_\theta})$ denote the corresponding global regularized distribution.*

The local consistency gap $\delta_{\boldsymbol{\theta},\Omega_\varepsilon}$ is bounded by the maximum probability mass of the global regularized distribution $\boldsymbol{p}_{\Omega_\varepsilon}(\boldsymbol{s_\theta})$ falling outside a local neighborhood:

$$\delta_{\boldsymbol{\theta},\Omega_\varepsilon} \leq \max_{\boldsymbol{y}^{(k)},S^{(k)}} \boldsymbol{p}_{\Omega_\varepsilon}(\boldsymbol{s_\theta})(\mathcal{Y} \setminus \mathcal{Y}^{(k)}),$$

where the maximum is taken over all feasible solutions $\boldsymbol{y}^{(k)} \in \mathcal{Y}$ and subsets of coordinates $S^{(k)} \subseteq [d]$, which define the neighborhood $\mathcal{Y}^{(k)}$.

The proof is given in Appendix B.11.

A.3 Constraint graph and cut-bounded bias

In this section, we analyze the bias of perturbed LNS (i.e., RLNS with perturbation-based regularization as defined in Section 3.3) through the lens of its variable selection scheme. We begin with an idealized scenario: when the feasible set perfectly factorizes into independent blocks of variables, perturbed LNS achieves zero bias if its variable selection scheme is aligned to this decomposition.

Proposition 11 (Exactness of perturbed LNS on product spaces)**.** *Let the coordinates $[d]$ be partitioned into M disjoint blocks $[d] = \bigsqcup_{m=1}^{M} B_m$, and assume the feasible set factorizes accordingly: $\mathcal{Y} = \prod_{m=1}^{M} \mathcal{Y}_m$, where $\mathcal{Y}_m \subset \mathbb{R}^{|B_m|}$.*

Assume that the noise distribution $\mathcal{D}$ factorizes over this partition, meaning that if $Z \sim \mathcal{D}$, the block subvectors $(Z_{[B_m]})_{m=1}^{M}$ are mutually independent.

If, at each step k, the frozen subset of coordinates $S^{(k)}$ is a union of blocks, i.e., if we have $S^{(k)} = \bigsqcup_{m \in \mathcal{M}^{(k)}} B_m$ for some index set $\mathcal{M}^{(k)} \subset [M]$, then $\delta_{\boldsymbol{\theta},\Omega_\varepsilon} = 0$, and Algorithm 3 implements an exact block Gibbs sampler, giving $\pi^{\mathrm{LNS}}_{\boldsymbol{\theta},\Omega_\varepsilon} = \boldsymbol{p}_{\Omega_\varepsilon}(\boldsymbol{s_\theta})$.

We give the proof in Appendix B.12. While perfect factorization guarantees zero bias, most combinatorial problems feature interacting variables. Typically, the feasible set is defined by m linear constraints $C \triangleq \{1, \ldots, m\}$:

$$\mathcal{Y} \triangleq \{\boldsymbol{y} \in \{0,1\}^d \mid \boldsymbol{A}\boldsymbol{y} \leq \boldsymbol{b}\},$$

where $\boldsymbol{A} \in \mathbb{R}^{m \times d}$ and $\boldsymbol{b} \in \mathbb{R}^m$. We analyze these interactions via the problem's constraint graph.

Definition 7 (Constraint hypergraph)**.** *The* constraint hypergraph *associated with the feasible set $\mathcal{Y}$ is a hypergraph $\mathcal{H}_C = (\mathcal{V}, \mathcal{E}_C)$, where the vertices correspond to the optimization variables $\mathcal{V} = [d]$, and each hyperedge $e_c \in \mathcal{E}_C$ corresponds to a constraint $c \in C$, containing exactly the variables involved in that constraint:*

$$\mathcal{E}_C \triangleq \{e_c \mid c \in C\} \quad \textit{where} \quad e_c \triangleq \{i \in [d] \mid A_{c,i} \neq 0\}.$$

At any iteration k, the choice of frozen variables $S^{(k)} \subseteq [d]$ and free variables $F^{(k)} \triangleq [d] \setminus S^{(k)}$ essentially partitions the vertices of this constraint hypergraph. This partition naturally divides the constraint indices C (and thus the hyperedges) into three disjoint sets:

- $C_F(S^{(k)})$: the constraints exclusively involving free variables in $F^{(k)}$;
- $C_S(S^{(k)})$: the constraints exclusively involving frozen variables in $S^{(k)}$;
- $C_\partial(S^{(k)})$: the *cut constraints*, which involve at least one variable in $F^{(k)}$ and one in $S^{(k)}$ (corresponding to the hyperedges in the cut-set of $\mathcal{H}_C$).

We can therefore decompose the constraints and write the cut constraints evaluated on a vector $\boldsymbol{y} \in \mathbb{R}^d$ as $\boldsymbol{A}_{[C_\partial(S^{(k)}),F^{(k)}]}\boldsymbol{y}_{[F^{(k)}]} + \boldsymbol{A}_{[C_\partial(S^{(k)}),S^{(k)}]}\boldsymbol{y}_{[S^{(k)}]} \leq \boldsymbol{b}_{[C_\partial(S^{(k)})]}$.

We now define a relaxed space where these cut constraints are dropped. This effectively forces the subproblem back into the idealized product-space setting described in Proposition 11.

Definition 8 (Relaxed product space)**.** *For a given partition defined by $S^{(k)}$, we define the* relaxed feasible set $\mathcal{Y}_{relax}(S^{(k)}) \triangleq \mathcal{Y}_F(S^{(k)}) \times \mathcal{Y}_S(S^{(k)})$*, where:*

$$\mathcal{Y}_F(S^{(k)}) \triangleq \left\{ \boldsymbol{y}_F \in \{0,1\}^{|F^{(k)}|} \mid \boldsymbol{A}_{[C_F(S^{(k)}),F^{(k)}]}\boldsymbol{y}_F \leq \boldsymbol{b}_{[C_F(S^{(k)})]} \right\},$$
$$\mathcal{Y}_S(S^{(k)}) \triangleq \left\{ \boldsymbol{y}_S \in \{0,1\}^{|S^{(k)}|} \mid \boldsymbol{A}_{[C_S(S^{(k)}),S^{(k)}]}\boldsymbol{y}_S \leq \boldsymbol{b}_{[C_S(S^{(k)})]} \right\}.$$

Notice that $\mathcal{Y} \subseteq \mathcal{Y}_{\text{relax}}(S^{(k)})$, and equality holds if and only if $C_\partial(S^{(k)}) = \emptyset$. Thus, the relaxed space strictly encompasses the true feasible set, allowing us to bound the local consistency gap by measuring how often the unconstrained, relaxed updates violate the dropped cut constraints.

Proposition 12 (Cut-bounded consistency gap)**.** *Assume the noise distribution $\mathcal{D}$ has mutually independent coordinates. For any frozen coordinates $S^{(k)} \subseteq [d]$ with $F^{(k)} \triangleq [d] \setminus S^{(k)}$, let $W_c(S^{(k)}) \subseteq \mathcal{Y}_{relax}(S^{(k)})$ be the set of relaxed solutions violating the cut constraint $c \in C_\partial(S^{(k)})$:*

$$W_c(S^{(k)}) \triangleq \left\{ \boldsymbol{y} \in \mathcal{Y}_{relax}(S^{(k)}) \mid \boldsymbol{A}_{[\{c\},F^{(k)}]}\boldsymbol{y}_{[F^{(k)}]} + \boldsymbol{A}_{[\{c\},S^{(k)}]}\boldsymbol{y}_{[S^{(k)}]} > b_c \right\},$$

and define the global and relaxed perturbed maximizers for $Z \sim \mathcal{D}$ as:

$$Y^* \triangleq \arg\max_{\boldsymbol{y} \in \mathcal{Y}} \langle \boldsymbol{\theta} + \varepsilon Z, \boldsymbol{y} \rangle, \quad \textit{and} \quad Y^*_{relax}(S^{(k)}) \triangleq \arg\max_{\boldsymbol{y} \in \mathcal{Y}_{relax}(S^{(k)})} \langle \boldsymbol{\theta} + \varepsilon Z, \boldsymbol{y} \rangle.$$

Denoting $W_\partial(S^{(k)}) \triangleq \bigcup_{c \in C_\partial(S^{(k)})} W_c(S^{(k)}) = \mathcal{Y}_{relax}(S^{(k)}) \setminus \mathcal{Y}$, we have:

$$\delta_{\boldsymbol{\theta},\Omega_\varepsilon} \leq \max_{\boldsymbol{y}^{(k)},S^{(k)}} \Big[\mathbb{P}\big(Y^*_{relax}(S^{(k)}) \in W_\partial(S^{(k)}) \;\big|\; (Y^*_{relax}(S^{(k)}))_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]}\big) + \mathbb{P}\big(Y^*_{relax}(S^{(k)}) \in W_\partial(S^{(k)}) \;\big|\; Y^*_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]}\big) \Big].$$

Furthermore, applying a union bound over $c \in C_\partial(S^{(k)})$ yields:

$$\delta_{\boldsymbol{\theta},\Omega_\varepsilon} \leq \max_{\boldsymbol{y}^{(k)},S^{(k)}} \Big\{ |C_\partial(S^{(k)})| \cdot \max_{c \in C_\partial(S^{(k)})} \Big[\mathbb{P}\big(Y^*_{relax}(S^{(k)}) \in W_c(S^{(k)}) \;\big|\; (Y^*_{relax}(S^{(k)}))_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]}\big) + \mathbb{P}\big(Y^*_{relax}(S^{(k)}) \in W_c(S^{(k)}) \;\big|\; Y^*_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]}\big) \Big] \Big\}.$$

Motivation for min-cut variable selection. Proposition 12, proved in Appendix B.13, provides a justification for constraint-graph heuristics in LNS. The upper bound on the bias of the local update scales with $|C_\partial(S^{(k)})|$, the number of cut constraints separating the frozen variables $S^{(k)}$ from the free variables $F^{(k)}$. This motivates the design of variable selection schemes that minimize $|C_\partial(S^{(k)})|$ in RLNS. In Appendix A.4, we empirically validate this theoretical insight on the maximum weight independent set problem.

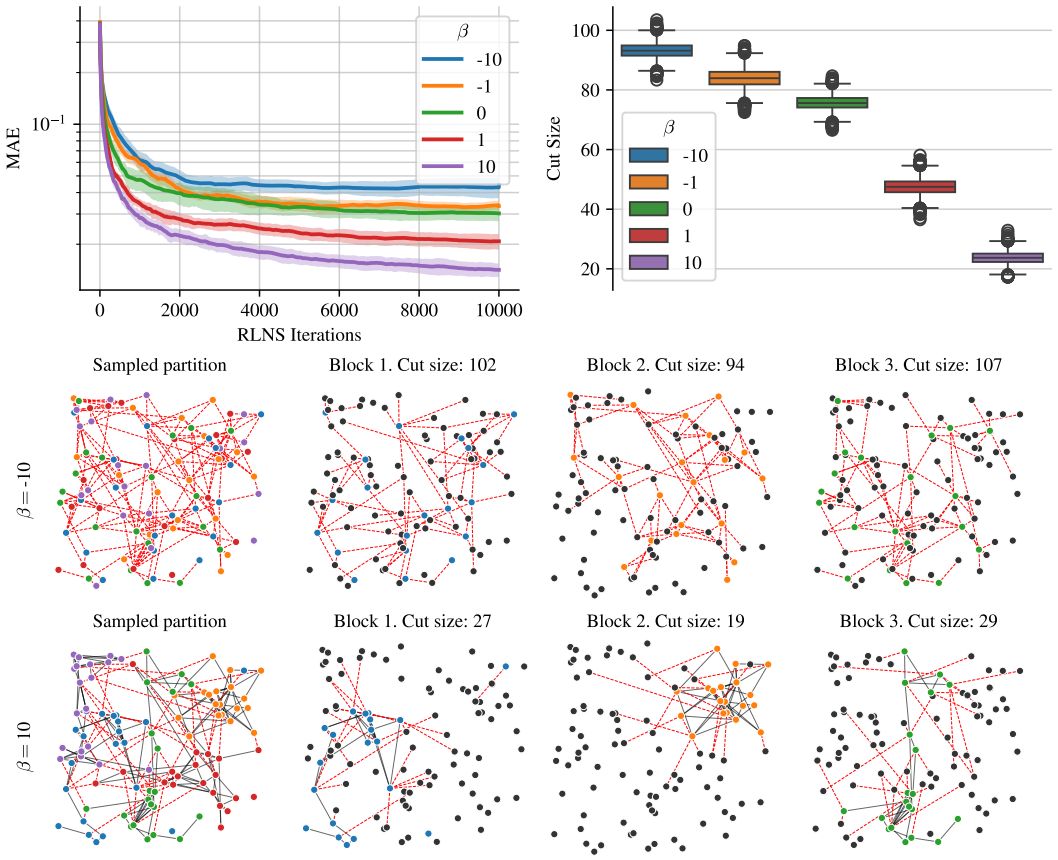


Figure 3: **Top left.** MAE between K-th RLNS estimate and true global expectation, computed as $\frac{1}{|\mathcal{N}|}\|\frac{1}{K}\sum_{k=1}^{K}\boldsymbol{y}^{(k)} - \mathbb{E}_{\boldsymbol{p}_{\Omega_\varepsilon}(\boldsymbol{s}_\theta)}[Y]\|_1$. **Top right.** Average cut size of the selected variables with different values of β. **Bottom.** Visualization of a sampled partition for $\beta = -10$ and $\beta = 10$. Internal edges are represented by black lines, and cut edges are represented by red dashed lines.

A.4 Min-cut RLNS on the maximum weight independent set

To empirically validate the cut-size-based bias bound from Appendix A.3, we evaluate RLNS with perturbation-based regularization on the maximum weight independent set (MWIS) problem. Given an undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, the MWIS problem writes:

$$\max_{\boldsymbol{y}\in\{0,1\}^{|\mathcal{N}|}} \sum_{n\in\mathcal{N}} \theta_n y_n$$
$$\text{s.t.} \quad y_u + y_v \leq 1 \quad \forall (u,v) \in \mathcal{E}.$$

Constraint graph. For the MWIS, the constraint hypergraph $\mathcal{H}_C$ exactly reduces to the standard problem graph $\mathcal{G}$: since each constraint involves exactly two nodes, every hyperedge has size 2 and corresponds directly to an edge in $\mathcal{G}$. Specifically, we generate soft random geometric graphs by randomly placing $|\mathcal{N}| = 100$ nodes in a 2D space and assigning edges with a probability that decays exponentially with the Euclidean distance between nodes. To test the impact of the cut size isolated in Proposition 12, we design variable selection schemes with varying average cut sizes.

Variable selection scheme. For each graph instance, we perform an offline phase where we generate a pool of 100 node partitions, which all divide $\mathcal{N}$ into 5 blocks of 20 nodes each. These partitions are generated via 10^4 steps of Metropolis-Hastings MCMC (where a proposal partition is obtained by exchanging the block labels of two random nodes) to be sampled approximately proportionally to

$\exp(-\beta \cdot c)$, where the cut size c is the total number of edges $(u, v) \in \mathcal{E}$ such that u and v belong to different blocks in the partition. By adjusting the inverse temperature β, we can smoothly interpolate the variable selection scheme from favoring max-cuts ($\beta < 0$) to min-cuts ($\beta > 0$): see Fig. 3 (bottom) for a visualization of sampled partitions. During the online RLNS procedure, at each step, we uniformly sample a partition from the offline pool and pick exactly one of its blocks uniformly at random to serve as the free variables, keeping the assignments of all other nodes frozen.

Results. We report results averaged across 10 random seeds, where each seed controls the generation of a distinct random graph instance. As illustrated in Fig. 3 (top), the results validate our theoretical insights from Appendix A.3: variable selection schemes with minimal cut size $|C_\partial(S^{(k)})|$ in the constraint graph yield a better convergence to the exact global regularized distribution $\boldsymbol{p}_{\Omega_\varepsilon}(\boldsymbol{s}_{\boldsymbol{\theta}})$.

B Proofs

B.1 Proof of Proposition 1 (Entropic LNS as a block Gibbs sampler)

Proof. When $\Omega = -\gamma H^s$, we have, for any $D \in \mathbb{N}$, $\boldsymbol{x} \in \mathbb{R}^D$:

$$\boldsymbol{p}_\Omega(\boldsymbol{x}) = \left(\frac{\exp(x_i/\gamma)}{\sum_{j=1}^D \exp(x_j/\gamma)} \right)_{i=1}^D.$$

Thus, on the one hand, the local regularized distribution writes, for all $\boldsymbol{y}^{(k)} \in \mathcal{Y}$ and $\boldsymbol{y} \in \mathcal{Y}^{(k)}$:

$$\boldsymbol{p}_{\Omega^{\mathcal{Y}^{(k)}}}\left((\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]}\right)(\boldsymbol{y}) = \frac{\exp\left(\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle / \gamma\right)}{\sum_{\boldsymbol{y}' \in \mathcal{Y}^{(k)}} \exp(\langle \boldsymbol{\theta}, \boldsymbol{y}' \rangle / \gamma)}.$$

On the other hand, the conditional probability of the global regularized distribution is given by:

$$\begin{aligned}
\boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})\left(\boldsymbol{y} \mid \boldsymbol{y}^{(k)}_{[S^{(k)}]}\right) &\triangleq \frac{\boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})(\boldsymbol{y})}{\sum\limits_{\substack{\boldsymbol{y}' \in \mathcal{Y} \\ \text{s.t.} \boldsymbol{y}'_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]}}} \boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})(\boldsymbol{y}')} \\
&= \frac{\boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})(\boldsymbol{y})}{\sum_{\boldsymbol{y}' \in \mathcal{Y}^{(k)}} \boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})(\boldsymbol{y}')} \\
&= \frac{\exp(\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle / \gamma)}{\sum_{\boldsymbol{y}'' \in \mathcal{Y}} \exp(\langle \boldsymbol{\theta}, \boldsymbol{y}'' \rangle / \gamma)} \times \left(\sum_{\boldsymbol{y}' \in \mathcal{Y}^{(k)}} \frac{\exp(\langle \boldsymbol{\theta}, \boldsymbol{y}' \rangle / \gamma)}{\sum_{\boldsymbol{y}'' \in \mathcal{Y}} \exp(\langle \boldsymbol{\theta}, \boldsymbol{y}'' \rangle / \gamma)} \right)^{-1} \\
&= \frac{\exp(\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle / \gamma)}{\sum_{\boldsymbol{y}'' \in \mathcal{Y}} \exp(\langle \boldsymbol{\theta}, \boldsymbol{y}'' \rangle / \gamma)} \times \frac{\sum_{\boldsymbol{y}'' \in \mathcal{Y}} \exp(\langle \boldsymbol{\theta}, \boldsymbol{y}'' \rangle / \gamma)}{\sum_{\boldsymbol{y}' \in \mathcal{Y}^{(k)}} \exp(\langle \boldsymbol{\theta}, \boldsymbol{y}' \rangle / \gamma)} \\
&= \frac{\exp(\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle / \gamma)}{\sum_{\boldsymbol{y}' \in \mathcal{Y}^{(k)}} \exp(\langle \boldsymbol{\theta}, \boldsymbol{y}' \rangle / \gamma)},
\end{aligned}$$

which therefore gives:

$$\boldsymbol{p}_{\Omega^{\mathcal{Y}^{(k)}}}\left((\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]}\right)(\boldsymbol{y}) = \boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})\left(\boldsymbol{y} \mid \boldsymbol{y}^{(k)}_{[S^{(k)}]}\right).$$

Thus, Algorithm 2 implements a block Gibbs sampler for the target distribution $\pi \triangleq \boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})$. To check that detailed balance is satisfied, let $\nu(S^{(k)})$ denote the probability of selecting the frozen subset $S^{(k)}$. The transition probability from $\boldsymbol{y}^{(k)}$ to $\boldsymbol{y}$ is:

$$P(\boldsymbol{y}^{(k)} \to \boldsymbol{y}) = \sum_{S^{(k)} \subseteq [d]} \nu(S^{(k)}) \mathbb{1}_{\{\boldsymbol{y}_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]}\}} \pi\left(\boldsymbol{y} \mid \boldsymbol{y}^{(k)}_{[S^{(k)}]}\right).$$

Multiplying by the stationary distribution candidate $\pi(\boldsymbol{y}^{(k)})$, we get:

$$\begin{aligned}\pi(\boldsymbol{y}^{(k)})P(\boldsymbol{y}^{(k)} \to \boldsymbol{y}) &= \sum_{S^{(k)} \subseteq [d]} \nu(S^{(k)}) \mathbb{1}_{\{\boldsymbol{y}_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]}\}} \pi(\boldsymbol{y}^{(k)}) \frac{\pi(\boldsymbol{y})}{\sum_{\boldsymbol{z} \in \mathcal{Y}^{(k)}} \pi(\boldsymbol{z})} \\ &= \sum_{S^{(k)} \subseteq [d]} \nu(S^{(k)}) \mathbb{1}_{\{\boldsymbol{y}_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]}\}} \frac{\pi(\boldsymbol{y}^{(k)})\pi(\boldsymbol{y})}{\sum_{\boldsymbol{z}:\, \boldsymbol{z}_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]}} \pi(\boldsymbol{z})}.\end{aligned}$$

Because the indicator function enforces $\boldsymbol{y}_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]}$, the denominator is strictly identical whether we condition on $\boldsymbol{y}^{(k)}_{[S^{(k)}]}$ or $\boldsymbol{y}_{[S^{(k)}]}$. The expression inside the sum is therefore perfectly symmetric with respect to $\boldsymbol{y}^{(k)}$ and $\boldsymbol{y}$, which immediately yields:

$$\pi(\boldsymbol{y}^{(k)})P(\boldsymbol{y}^{(k)} \to \boldsymbol{y}) = \pi(\boldsymbol{y})P(\boldsymbol{y} \to \boldsymbol{y}^{(k)}),$$

so that $\boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})$ is a stationary distribution of the Markov chain. Finally, an irreducible variable selection scheme ensures the chain is ergodic (since the strictly positive entropic update distributions guarantee dense transitions and aperiodicity via self-loops). This gives that the stationary distribution is uniquely equal to $\pi^{\mathrm{LNS}}_{\boldsymbol{\theta},\Omega} = \boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})$, and that the chain converges to it. □

B.2 Proof of Proposition 2 (Characterization of stationary exactness)

Proof. The sense ($\Longleftarrow$) is established by Proposition 1, so we focus on the ($\Longrightarrow$) implication.

Assume that $\Omega : \triangle^{\mathcal{Y}} \to \mathbb{R}$ is separable and Legendre-type on $\triangle^{\mathcal{Y}}$ [Wainwright and Jordan, 2008, Blondel et al., 2020], meaning $\Omega(\boldsymbol{p}) = \sum_{\boldsymbol{y} \in \mathcal{Y}} \omega(p_{\boldsymbol{y}})$ for some l.s.c. strictly convex function $\omega : [0,1] \to \mathbb{R}$ that is differentiable on $(0,1)$ with $\lim_{p \to 0^+} \omega'(p) = -\infty$ (so that $\boldsymbol{p}_\Omega$ maps $\mathbb{R}^{|\mathcal{Y}|}$ onto $\mathrm{ri}(\triangle^{\mathcal{Y}})$ and local update probabilities are strictly positive). Since $\Omega(\delta^{\mathcal{Y}}_{\boldsymbol{y}}) = 0 \;\; \forall \boldsymbol{y} \in \mathcal{Y}$, we assume standard simplex normalization $\omega(0) = \omega(1) = 0$ without loss of generality. We assume that $\pi^{\mathrm{LNS}}_{\boldsymbol{\theta},\Omega} = \boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})$ for all score vectors $\boldsymbol{s}_{\boldsymbol{\theta}}$, and we will show this forces $\Omega = -\gamma H^s$ for some $\gamma > 0$.

Problem definition. Consider a state space of dimension $d = 3$ with feasible set $\mathcal{Y} = \{\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3\}$. Let $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3) \in \mathbb{R}^3$ be any maximization direction. Note that in this setting, we have $\boldsymbol{s}_{\boldsymbol{\theta}} = \boldsymbol{\theta}$. We instantiate the following specific variable selection strategy for Algorithm 2: at each step k, given a current solution $\boldsymbol{y}^{(k)} = \boldsymbol{e}_i$, the algorithm samples a frozen coordinate subset $S^{(k)} \in \{\{1\}, \{3\}\}$ uniformly at random. We strictly prevent freezing coordinate 2.

This choice restricts the possible transitions. For instance, if $\boldsymbol{y}^{(k)} = \boldsymbol{e}_1$ (meaning $y^{(k)}_1 = 1$), freezing coordinate 3 (where $y^{(k)}_3 = 0$) creates the local neighborhood $\mathcal{Y}^{(k)} = \{\boldsymbol{e}_1, \boldsymbol{e}_2\}$. However, it is impossible to transition directly between $\boldsymbol{e}_1$ and $\boldsymbol{e}_3$ in a single step, as that would require freezing coordinate 2, which our algorithm never does. The state transition graph on $\mathcal{Y}$ is therefore a simple line graph: $\boldsymbol{e}_1 \leftrightarrow \boldsymbol{e}_2 \leftrightarrow \boldsymbol{e}_3$.

Detailed balance. Because the transition graph is a tree (it contains no cycles), any stationary distribution on this graph must satisfy detailed balance. Since we assume the RLNS stationary distribution matches the global target ($\pi^{\mathrm{LNS}}_{\boldsymbol{\theta},\Omega} = \boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})$), the target distribution $\boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})$ itself must satisfy detailed balance along the edges of the tree. Let $\boldsymbol{\pi} \triangleq \boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})$, i.e., $\pi_i \triangleq \boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})(\boldsymbol{e}_i)$. The balance equation between $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ is:

$$\pi_1 \cdot \mathbb{P}(\boldsymbol{e}_1 \to \boldsymbol{e}_2) = \pi_2 \cdot \mathbb{P}(\boldsymbol{e}_2 \to \boldsymbol{e}_1).$$

The transition from $\boldsymbol{e}_1$ to $\boldsymbol{e}_2$ (resp. from $\boldsymbol{e}_2$ to $\boldsymbol{e}_1$) occurs by choosing to freeze coordinate 3 (which occurs with probability $1/2$) and then sampling $\boldsymbol{e}_2$ (resp. $\boldsymbol{e}_1$) from the local distribution $q^{\{1,2\}} \triangleq \boldsymbol{p}_{\Omega^{\mathcal{Y}^{(k)}}}((\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]})$ (where $\mathcal{Y}^{(k)} = \{\boldsymbol{e}_1, \boldsymbol{e}_2\}$). To simplify notation, let $q_1 \triangleq q^{\{1,2\}}(\boldsymbol{e}_1)$ and $q_2 \triangleq q^{\{1,2\}}(\boldsymbol{e}_2)$. We have:

$$\pi_1 \cdot \left(\frac{1}{2} q_2\right) = \pi_2 \cdot \left(\frac{1}{2} q_1\right).$$

Since local probabilities sum to 1, substituting $q_2 = 1 - q_1$ immediately yields:

$$q_1 = \frac{\pi_1}{\pi_1 + \pi_2} \quad \text{and} \quad q_2 = \frac{\pi_2}{\pi_1 + \pi_2}. \tag{3}$$

Optimality conditions. Because $\Omega(\boldsymbol{p}) = \sum_{i=1}^{3} \omega(p_i)$ is strictly convex and differentiable on the interior of the simplex, the global distribution $\boldsymbol{\pi} = \boldsymbol{p}_\Omega(\boldsymbol{s_\theta})$ is by definition the unique solution to the following constrained optimization problem:

$$\begin{aligned} \min_{\boldsymbol{p}>0} \quad & \sum_{i=1}^{3} \big(\omega(p_i) - \theta_i p_i\big) \\ \text{s.t.} \quad & \sum_{i=1}^{3} p_i = 1. \end{aligned}$$

We introduce a Lagrange multiplier $\nu \in \mathbb{R}$ for the simplex equality constraint. The Lagrangian is given by:

$$\mathcal{L}(\boldsymbol{p}, \nu) = \sum_{i=1}^{3} \big(\omega(p_i) - \theta_i p_i\big) - \nu \left(\sum_{i=1}^{3} p_i - 1 \right).$$

The Karush-Kuhn-Tucker (KKT) stationarity conditions require the partial derivatives of the Lagrangian with respect to each p_i to be zero at optimality:

$$\frac{\partial \mathcal{L}}{\partial p_i}(\boldsymbol{\pi}, \nu^\star) = \omega'(\pi_i) - \theta_i - \nu^\star = 0 \implies \omega'(\pi_i) = \theta_i + \nu^\star.$$

By subtracting the optimality condition of $i = 2$ from $i = 1$, the multiplier $\nu^\star$ cancels out, yielding:

$$\omega'(\pi_1) - \omega'(\pi_2) = \theta_1 - \theta_2. \tag{4}$$

Similarly, the local distribution $q^{\{1,2\}}$ solves by definition the local constrained problem:

$$\begin{aligned} \min_{q_1, q_2 > 0} \quad & \big(\omega(q_1) - \theta_1 q_1\big) + \big(\omega(q_2) - \theta_2 q_2\big) \\ \text{s.t.} \quad & q_1 + q_2 = 1. \end{aligned}$$

Note that the $\omega(0)$ term due to padding the local distribution $\boldsymbol{q}$ into a global one (see the definition of the local regularizer in Definition 2) vanishes since we assumed $\omega(0) = \omega(1) = 0$. Introducing a local Lagrange multiplier $\mu \in \mathbb{R}$, the local Lagrangian is:

$$\mathcal{L}_{\text{local}}(\boldsymbol{q}, \mu) = \big(\omega(q_1) - \theta_1 q_1\big) + \big(\omega(q_2) - \theta_2 q_2\big) - \mu(q_1 + q_2 - 1).$$

The KKT stationarity conditions for the local problem similarly yield $\omega'(q_i) = \theta_i + \mu^\star$. Subtracting the condition for q_2 from q_1 gives:

$$\omega'(q_1) - \omega'(q_2) = \theta_1 - \theta_2. \tag{5}$$

Combining Eq. (4) and Eq. (5), we get:

$$\omega'(\pi_1) - \omega'(\pi_2) = \omega'(q_1) - \omega'(q_2).$$

Finally, substituting the expressions of q_1 and q_2 derived in Eq. (3), we get:

$$\omega'(\pi_1) - \omega'(\pi_2) = \omega'\left(\frac{\pi_1}{\pi_1 + \pi_2}\right) - \omega'\left(\frac{\pi_2}{\pi_1 + \pi_2}\right). \tag{6}$$

Functional equation. Let $y \triangleq \pi_1 + \pi_2$ and $x \triangleq \frac{\pi_1}{\pi_1 + \pi_2}$, such that $\pi_1 = xy$ and $\pi_2 = (1-x)y$. Since the global scores $\theta_1, \theta_2, \theta_3$ are unconstrained in $\mathbb{R}$ and $\boldsymbol{p}_\Omega$ is surjective onto the relative interior $\operatorname{ri}(\triangle^{\mathcal{Y}})$ (as $\boldsymbol{p}_\Omega = \nabla\Omega^*$ and Ω is Legendre-type), $\boldsymbol{\pi}$ spans $\operatorname{ri}(\triangle^{\mathcal{Y}})$, so we have that (π_1, π_2) spans $\{(\pi_1, \pi_2) \in (0,1)^2 \mid \pi_1 + \pi_2 < 1\}$ and thus x and y independently span $(0, 1)$.

Substituting $\pi_1 = xy$ and $\pi_2 = (1-x)y$ into Eq. (6) yields:

$$\omega'(xy) - \omega'((1-x)y) = \omega'(x) - \omega'(1-x).$$

We integrate this equation with respect to x (holding y constant):

$$\begin{aligned} & \int \left[\omega'(xy) - \omega'((1-x)y)\right] dx = \int \left[\omega'(x) - \omega'(1-x)\right] dx \\ \implies & \frac{1}{y}\omega(xy) + \frac{1}{y}\omega((1-x)y) = \omega(x) + \omega(1-x) + C(y). \end{aligned}$$

Multiplying by y, we get:

$$\omega(xy) + \omega((1-x)y) = y(\omega(x) + \omega(1-x)) + yC(y). \tag{7}$$

To determine the integration term $yC(y)$, we evaluate the limit as $x \to 1$. Because we normalized $\omega(0) = \omega(1) = 0$, taking $x \to 1$ yields:

$$\omega(y) + \omega(0) = y(\omega(1) + \omega(0)) + yC(y) \implies \omega(y) = yC(y).$$

Substituting $yC(y) = \omega(y)$ back into Eq. (7) yields the functional equation:

$$\omega(xy) + \omega((1-x)y) - \omega(y) = y(\omega(x) + \omega(1-x)) \quad \forall\, x, y \in (0,1).$$

As first established in Horibe [1988, Theorem 0] and extended in Gselmann [2011], all measurable solutions to this functional equation take the form $\omega(p) = \gamma p \log p + cp$ for some constants $\gamma, c \in \mathbb{R}$. Since we assumed $\omega(1) = 0$, we have $c = 0$. Moreover, since we assumed ω to be strictly convex, we get $\gamma > 0$, which concludes the proof. □

B.3 Strict convexity of Ω_ε

In this section, we establish that the perturbation-based regularizer Ω_ε defined in Definition 4 is strictly convex over the entire closed simplex $\triangle^{\mathcal{Y}}$.

Proposition 13 (Strict convexity of Ω_ε)**.** *Let $\Omega_\varepsilon : \triangle^{\mathcal{Y}} \to \mathbb{R}$ be the perturbation-based regularization function defined in Definition 4. If the noise distribution $\mathcal{D}$ admits a strictly positive density on $\mathbb{R}^d$, then Ω_ε is strictly convex on the entire closed simplex $\triangle^{\mathcal{Y}}$.*

Proof. Let $\boldsymbol{p}, \boldsymbol{q} \in \triangle^{\mathcal{Y}}$ be two distinct probability distributions, and let $\lambda \in (0,1)$. We define their convex combination $\boldsymbol{r} \triangleq \lambda \boldsymbol{p} + (1-\lambda)\boldsymbol{q}$. The support of $\boldsymbol{r}$ is the union of the supports of $\boldsymbol{p}$ and $\boldsymbol{q}$, i.e., $\mathcal{Z} \triangleq \operatorname{supp}(\boldsymbol{r}) = \operatorname{supp}(\boldsymbol{p}) \cup \operatorname{supp}(\boldsymbol{q}) \subseteq \mathcal{Y}$.

Consider the local simplex $\triangle^{\mathcal{Z}}$ and the inclusion map $\iota_{\mathcal{Z}} : \triangle^{\mathcal{Z}} \hookrightarrow \triangle^{\mathcal{Y}}$ as defined in Definition 2. By construction, $\boldsymbol{p}$ and $\boldsymbol{q}$ belong to the face $\iota_{\mathcal{Z}}(\triangle^{\mathcal{Z}})$, and $\boldsymbol{r}$ lies strictly in the relative interior of this face, meaning $\boldsymbol{r} \in \operatorname{ri}(\iota_{\mathcal{Z}}(\triangle^{\mathcal{Z}}))$.

To prove strict convexity, we must show that:

$$\Omega_\varepsilon(\boldsymbol{r}) < \lambda \Omega_\varepsilon(\boldsymbol{p}) + (1-\lambda)\Omega_\varepsilon(\boldsymbol{q}).$$

Assume for contradiction that equality holds. Since Ω_ε is a supremum of affine functions, it is convex and lower semi-continuous. The equality at an interior point of the segment $[\boldsymbol{p}, \boldsymbol{q}]$ implies that Ω_ε must be strictly affine on the entire segment $[\boldsymbol{p}, \boldsymbol{q}]$. Because $\boldsymbol{r} \in \operatorname{ri}(\iota_{\mathcal{Z}}(\triangle^{\mathcal{Z}}))$, there exists a small line segment centered at $\boldsymbol{r}$, contained entirely within $[\boldsymbol{p}, \boldsymbol{q}]$, that lies completely within the relative interior $\operatorname{ri}(\iota_{\mathcal{Z}}(\triangle^{\mathcal{Z}}))$. If Ω_ε is affine on $[\boldsymbol{p}, \boldsymbol{q}]$, it must be affine on this sub-segment.

We now analyze the restriction of Ω_ε to this face. As shown in Appendix B.4, the local regularizer $\Omega_\varepsilon^{\mathcal{Z}} \triangleq \Omega_\varepsilon \circ \iota_{\mathcal{Z}}$ is exactly the Fenchel conjugate of the local perturbed maximum function $F_{\varepsilon,\mathcal{Z}} : \mathbb{R}^{|\mathcal{Z}|} \to \mathbb{R}$:

$$F_{\varepsilon,\mathcal{Z}}(\boldsymbol{u}) \triangleq \mathbb{E}_{Z \sim \mathcal{D}}\left[\max_{\boldsymbol{y} \in \mathcal{Z}} \{u_{\boldsymbol{y}} + \varepsilon \langle \boldsymbol{y}, Z \rangle\}\right].$$

By Bouvier et al. [2025, Theorem 4], we have that:

1. The domain of the conjugate of $F_{\varepsilon,\mathcal{Z}}$ is $\triangle^{\mathcal{Z}}$, i.e., $\operatorname{dom}(\Omega_\varepsilon^{\mathcal{Z}}) = \triangle^{\mathcal{Z}}$.
2. $F_{\varepsilon,\mathcal{Z}}$ is l.s.c., proper, convex and differentiable on $\mathbb{R}^{|\mathcal{Z}|}$, which makes it essentially smooth.

Rockafellar [1970, Theorem 26.3] then gives that its Fenchel conjugate $\Omega_\varepsilon^{\mathcal{Z}}$ is essentially strictly convex. Therefore, it is strictly convex on the relative interior of its domain, $\operatorname{ri}(\operatorname{dom}(\Omega_\varepsilon^{\mathcal{Z}})) = \operatorname{ri}(\triangle^{\mathcal{Z}})$.

This establishes that Ω_ε is strictly convex on $\operatorname{ri}(\iota_{\mathcal{Z}}(\triangle^{\mathcal{Z}}))$. However, we previously deduced that Ω_ε would be affine on a sub-segment passing through $\boldsymbol{r}$ within this relative interior, which is a contradiction. Therefore, the strict inequality must hold, concluding the proof that Ω_ε is strictly convex on the entire closed simplex $\triangle^{\mathcal{Y}}$. □

B.4 Proof of Proposition 3 (Perturbed LNS sampling)

Proof. Let $F_{\varepsilon,\mathcal{Y}^{(k)}}$ denote the following local perturbed maximum function, defined on scores restricted to the neighborhood $\mathcal{Y}^{(k)}$:

$$\begin{aligned} F_{\varepsilon,\mathcal{Y}^{(k)}} : \mathbb{R}^{|\mathcal{Y}^{(k)}|} &\longrightarrow \mathbb{R} \\ \boldsymbol{u} &\longmapsto \mathbb{E}_{Z\sim\mathcal{D}}\left[\max_{\boldsymbol{y}\in\mathcal{Y}^{(k)}} \{u_{\boldsymbol{y}} + \varepsilon\langle \boldsymbol{y}, Z\rangle\}\right]. \end{aligned}$$

We first establish that the local regularization function $\Omega_\varepsilon^{\mathcal{Y}^{(k)}} \triangleq \Omega_\varepsilon \circ \iota_{\mathcal{Y}^{(k)}}$ is the Fenchel conjugate of $F_{\varepsilon,\mathcal{Y}^{(k)}}$. Starting from the definition of Ω_ε, we have for any local distribution $\boldsymbol{q} \in \triangle^{\mathcal{Y}^{(k)}}$:

$$\begin{aligned} \Omega_\varepsilon^{\mathcal{Y}^{(k)}}(\boldsymbol{q}) &= \Omega_\varepsilon(\iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q})) \\ &= \sup_{\boldsymbol{s}\in\mathbb{R}^{|\mathcal{Y}|}} \left\{ \langle \boldsymbol{s}, \iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q})\rangle - \mathbb{E}_{Z\sim\mathcal{D}}\left[\max_{\boldsymbol{y}'\in\mathcal{Y}} \{s_{\boldsymbol{y}'} + \varepsilon\langle \boldsymbol{y}', Z\rangle\}\right]\right\} \\ &= \sup_{\boldsymbol{s}\in\mathbb{R}^{|\mathcal{Y}|}} \left\{ \left(\sum_{\boldsymbol{y}\in\mathcal{Y}^{(k)}} s_{\boldsymbol{y}} q_{\boldsymbol{y}}\right) - \mathbb{E}_{Z\sim\mathcal{D}}\left[\max_{\boldsymbol{y}'\in\mathcal{Y}} \{s_{\boldsymbol{y}'} + \varepsilon\langle \boldsymbol{y}', Z\rangle\}\right]\right\}. \end{aligned}$$

We decompose the global score vector $\boldsymbol{s}$ into components inside $\mathcal{Y}^{(k)}$ (denoted $\boldsymbol{s}_{\text{in}}$) and outside $\mathcal{Y}^{(k)}$ (denoted $\boldsymbol{s}_{\text{out}}$). To maximize the objective inside the supremum, we observe that $\boldsymbol{s}_{\text{out}}$ only appears in the negative expectation term. Sending the coordinates of $\boldsymbol{s}_{\text{out}}$ to $-\infty$ minimizes the expectation term (by eliminating $\boldsymbol{y}' \notin \mathcal{Y}^{(k)}$ from the maximum) without affecting the linear term $\langle \boldsymbol{s}_{\text{in}}, \boldsymbol{q}\rangle$.

Formally, we have:

$$\begin{aligned} \lim_{\substack{s_{\boldsymbol{y}'}\to-\infty \\ \forall \boldsymbol{y}'\in\mathcal{Y}\setminus\mathcal{Y}^{(k)}}} \mathbb{E}_{Z\sim\mathcal{D}}\left[\max_{\boldsymbol{y}\in\mathcal{Y}} \{s_{\boldsymbol{y}} + \varepsilon\langle \boldsymbol{y}, Z\rangle\}\right] &= \mathbb{E}_{Z\sim\mathcal{D}}\left[\max_{\boldsymbol{y}\in\mathcal{Y}^{(k)}} \{(\boldsymbol{s}_{\text{in}})_{\boldsymbol{y}} + \varepsilon\langle \boldsymbol{y}, Z\rangle\}\right] \\ &\leq \mathbb{E}_{Z\sim\mathcal{D}}\left[\max_{\boldsymbol{y}\in\mathcal{Y}} \{s_{\boldsymbol{y}} + \varepsilon\langle \boldsymbol{y}, Z\rangle\}\right] \ \ \forall \boldsymbol{s} \in \mathbb{R}^{|\mathcal{Y}|}. \end{aligned}$$

Substituting this back into the conjugate expression:

$$\begin{aligned} \Omega_\varepsilon^{\mathcal{Y}^{(k)}}(\boldsymbol{q}) &= \sup_{\boldsymbol{s}_{\text{in}}\in\mathbb{R}^{|\mathcal{Y}^{(k)}|}} \left\{ \langle \boldsymbol{s}_{\text{in}}, \boldsymbol{q}\rangle - \mathbb{E}_{Z\sim\mathcal{D}}\left[\max_{\boldsymbol{y}\in\mathcal{Y}^{(k)}} \{(\boldsymbol{s}_{\text{in}})_{\boldsymbol{y}} + \varepsilon\langle \boldsymbol{y}, Z\rangle\}\right]\right\} \\ &= \sup_{\boldsymbol{u}\in\mathbb{R}^{|\mathcal{Y}^{(k)}|}} \left\{ \langle \boldsymbol{u}, \boldsymbol{q}\rangle - F_{\varepsilon,\mathcal{Y}^{(k)}}(\boldsymbol{u})\right\} \\ &= F^*_{\varepsilon,\mathcal{Y}^{(k)}}(\boldsymbol{q}). \end{aligned}$$

Thus, the local regularizer is exactly the conjugate of the local perturbed maximum function.

Finally, we compute the local regularized distribution. By definition:

$$\begin{aligned} \boldsymbol{p}_{\Omega_\varepsilon^{\mathcal{Y}^{(k)}}}\left((\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]}\right) &\triangleq \arg\max_{\boldsymbol{q}\in\triangle^{\mathcal{Y}^{(k)}}} \left\{ \langle (\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]}, \boldsymbol{q}\rangle - \Omega_\varepsilon^{\mathcal{Y}^{(k)}}(\boldsymbol{q})\right\} \\ &= \arg\max_{\boldsymbol{q}\in\triangle^{\mathcal{Y}^{(k)}}} \left\{ \langle (\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]}, \boldsymbol{q}\rangle - F^*_{\varepsilon,\mathcal{Y}^{(k)}}(\boldsymbol{q})\right\}, \end{aligned}$$

so that Danskin's theorem gives:

$$\boldsymbol{p}_{\Omega_\varepsilon^{\mathcal{Y}^{(k)}}}\left((\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]}\right) = \nabla F^{**}_{\varepsilon,\mathcal{Y}^{(k)}}\left((\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]}\right).$$

Since $F_{\varepsilon,\mathcal{Y}^{(k)}}$ is closed proper convex, it is equal to its biconjugate by the Fenchel-Moreau theorem, and we have in fact:

$$\boldsymbol{p}_{\Omega_\varepsilon^{\mathcal{Y}^{(k)}}}\left((\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]}\right) = \nabla F_{\varepsilon,\mathcal{Y}^{(k)}}\left((\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]}\right).$$

We then invert the differentiation and expectation signs (this application of the *path-wise gradient estimator*, or the *reparameterization trick*, is possible since the expectation is taken with respect to Z

which does not depend on $\boldsymbol{s_\theta}$ [Blondel and Roulet, 2025]), and further apply Danskin's theorem a second time:

$$\begin{aligned}\nabla F_{\varepsilon,\mathcal{Y}^{(k)}}\left((\boldsymbol{s_\theta})_{[\mathcal{Y}^{(k)}]}\right) &= \nabla_{(\boldsymbol{s_\theta})_{[\mathcal{Y}^{(k)}]}} \mathbb{E}_{Z\sim\mathcal{D}}\left[\max_{\boldsymbol{y}\in\mathcal{Y}^{(k)}}\left\{\left((\boldsymbol{s_\theta})_{[\mathcal{Y}^{(k)}]}\right)_{\boldsymbol{y}} + \varepsilon\langle \boldsymbol{y}, Z\rangle\right\}\right] \\ &= \mathbb{E}_{Z\sim\mathcal{D}}\left[\nabla_{(\boldsymbol{s_\theta})_{[\mathcal{Y}^{(k)}]}} \max_{\boldsymbol{y}\in\mathcal{Y}^{(k)}}\left\{\left((\boldsymbol{s_\theta})_{[\mathcal{Y}^{(k)}]}\right)_{\boldsymbol{y}} + \varepsilon\langle \boldsymbol{y}, Z\rangle\right\}\right] \\ &= \mathbb{E}_{Z\sim\mathcal{D}}\left[\delta^{\mathcal{Y}^{(k)}}_{\arg\max_{\boldsymbol{y}\in\mathcal{Y}^{(k)}}\langle\boldsymbol{\theta}+\varepsilon Z,\boldsymbol{y}\rangle}\right].\end{aligned}$$

□

B.5 Proof of Proposition 7 (Expected RLNS as block-coordinate ascent)

Proof. We first establish the strict convexity of the mean-space regularizer $\Omega_{\mathcal{M}}$. Let $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \mathcal{M}$ such that $\boldsymbol{\mu}_1 \neq \boldsymbol{\mu}_2$, and let $\lambda \in (0,1)$. Because the simplex $\triangle^{\mathcal{Y}}$ is compact and Ω is strictly convex, the minimum in the definition of $\Omega_{\mathcal{M}}$ is uniquely attained for any $\boldsymbol{\mu} \in \mathcal{M}$. Let $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ be the unique minimizers corresponding to $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$, respectively. Since $\boldsymbol{\mu}_1 \neq \boldsymbol{\mu}_2$, we must have $\boldsymbol{p}_1 \neq \boldsymbol{p}_2$.

Define the convex combinations $\boldsymbol{\mu}_\lambda \triangleq \lambda\boldsymbol{\mu}_1 + (1-\lambda)\boldsymbol{\mu}_2$ and $\boldsymbol{p}_\lambda \triangleq \lambda\boldsymbol{p}_1 + (1-\lambda)\boldsymbol{p}_2$. By linearity of expectation, we have $\mathbb{E}_{\boldsymbol{p}_\lambda}[Y] = \boldsymbol{\mu}_\lambda$. Since $\boldsymbol{p}_\lambda \in \triangle^{\mathcal{Y}}$, it is a valid candidate distribution for the constrained minimization problem defining $\Omega_{\mathcal{M}}(\boldsymbol{\mu}_\lambda)$. Therefore, we have $\Omega_{\mathcal{M}}(\boldsymbol{\mu}_\lambda) \leq \Omega(\boldsymbol{p}_\lambda)$. Because Ω is strictly convex and $\boldsymbol{p}_1 \neq \boldsymbol{p}_2$, we strictly bound the right-hand side:

$$\Omega(\boldsymbol{p}_\lambda) < \lambda\Omega(\boldsymbol{p}_1) + (1-\lambda)\Omega(\boldsymbol{p}_2).$$

Substituting the definitions of the marginal minima yields $\Omega_{\mathcal{M}}(\boldsymbol{\mu}_\lambda) < \lambda\Omega_{\mathcal{M}}(\boldsymbol{\mu}_1) + (1-\lambda)\Omega_{\mathcal{M}}(\boldsymbol{\mu}_2)$, which proves that $\Omega_{\mathcal{M}}$ is strictly convex on $\mathcal{M}$.

We now turn to the BCA step. Let $\mathcal{M}^{(k)} \triangleq \{\boldsymbol{\mu} \in \mathcal{M} \mid \boldsymbol{\mu}_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]}\}$ denote the feasible set of the mean-space BCA step. We must show that this slice of the marginal polytope strictly corresponds to the convex hull of the local neighborhood $\mathcal{Y}^{(k)} \triangleq \{\boldsymbol{y} \in \mathcal{Y} \mid \boldsymbol{y}_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]}\}$.

Clearly, $\operatorname{conv}(\mathcal{Y}^{(k)}) \subseteq \mathcal{M}^{(k)}$. We must prove $\mathcal{M}^{(k)} \subseteq \operatorname{conv}(\mathcal{Y}^{(k)})$. Let $\boldsymbol{\mu} \in \mathcal{M}^{(k)}$. Because $\boldsymbol{\mu} \in \mathcal{M}$, there exists a distribution $\boldsymbol{p} \in \triangle^{\mathcal{Y}}$ such that $\boldsymbol{\mu} = \sum_{\boldsymbol{y}\in\mathcal{Y}} p_{\boldsymbol{y}}\boldsymbol{y}$. For any coordinate $i \in S^{(k)}$, we are given that $\mathcal{Y} \subseteq \{0,1\}^d$, so $y_i \in \{0,1\}$ for all $\boldsymbol{y} \in \mathcal{Y}$, and $y_i^{(k)} \in \{0,1\}$. Since $\boldsymbol{\mu} \in \mathcal{M}^{(k)}$, we have $\mu_i = y_i^{(k)}$. We distinguish two cases:

- If $y_i^{(k)} = 0$, then $\mu_i = 0$, giving $\sum_{\boldsymbol{y}\in\mathcal{Y}} p_{\boldsymbol{y}} y_i = 0$. Since $p_{\boldsymbol{y}} \geq 0$ and $y_i \in \{0,1\}$, all terms in the sum are non-negative. This implies that for any $\boldsymbol{y} \in \mathcal{Y}$, $p_{\boldsymbol{y}} > 0 \implies y_i = 0$.
- If $y_i^{(k)} = 1$, then $\mu_i = 1$, giving $\sum_{\boldsymbol{y}\in\mathcal{Y}} p_{\boldsymbol{y}} y_i = 1$. Since $\sum_{\boldsymbol{y}\in\mathcal{Y}} p_{\boldsymbol{y}} = 1$, we can rewrite this as $\sum_{\boldsymbol{y}\in\mathcal{Y}} p_{\boldsymbol{y}}(1-y_i) = 0$. Since $p_{\boldsymbol{y}} \geq 0$ and $1 - y_i \geq 0$, we similarly conclude that $p_{\boldsymbol{y}} > 0 \implies y_i = 1$.

In both cases, any solution $\boldsymbol{y}$ with non-zero probability mass ($p_{\boldsymbol{y}} > 0$) must strictly match $\boldsymbol{y}^{(k)}$ on all coordinates $i \in S^{(k)}$. Thus, $\operatorname{supp}(\boldsymbol{p}) \subseteq \mathcal{Y}^{(k)}$, which proves that $\boldsymbol{\mu} \in \operatorname{conv}(\mathcal{Y}^{(k)})$, and therefore $\mathcal{M}^{(k)} = \operatorname{conv}(\mathcal{Y}^{(k)})$.

Substituting the definition of the mean-space regularizer $\Omega_{\mathcal{M}}$ into the exact BCA update, the next mean-space iterate $\boldsymbol{\mu}^{(k+1)}$ is given by:

$$\boldsymbol{\mu}^{(k+1)} = \arg\max_{\boldsymbol{\mu}\in\mathcal{M}^{(k)}}\left[\langle\boldsymbol{\theta},\boldsymbol{\mu}\rangle - \min_{\boldsymbol{p}\in\triangle^{\mathcal{Y}}\text{ s.t. }\mathbb{E}_{\boldsymbol{p}}[Y]=\boldsymbol{\mu}} \Omega(\boldsymbol{p})\right].$$

Distributing the minus sign turns the inner minimum into a maximum:

$$\boldsymbol{\mu}^{(k+1)} = \arg\max_{\boldsymbol{\mu}\in\mathcal{M}^{(k)}}\left[\max_{\boldsymbol{p}\in\triangle^{\mathcal{Y}}\text{ s.t. }\mathbb{E}_{\boldsymbol{p}}[Y]=\boldsymbol{\mu}} \left(\langle\boldsymbol{\theta},\boldsymbol{\mu}\rangle - \Omega(\boldsymbol{p})\right)\right].$$

Using the fact that $\langle \boldsymbol{\theta}, \boldsymbol{\mu} \rangle = \langle \boldsymbol{\theta}, \mathbb{E}_{\boldsymbol{p}}[Y] \rangle = \langle \boldsymbol{s}_{\boldsymbol{\theta}}, \boldsymbol{p} \rangle$ via the definition of the score vector $\boldsymbol{s}_{\boldsymbol{\theta}} \triangleq (\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle)_{\boldsymbol{y} \in \mathcal{Y}}$, we can absorb the maximization over $\boldsymbol{\mu}$ and directly optimize over the distribution $\boldsymbol{p}$:

$$\boldsymbol{\mu}^{(k+1)} = \mathbb{E}_{\boldsymbol{p}^*}[Y], \quad \text{where} \quad \boldsymbol{p}^* = \underset{\substack{\boldsymbol{p} \in \triangle^{\mathcal{Y}} \\ \mathbb{E}_{\boldsymbol{p}}[Y]_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]}}}{\arg\max} \left[\langle \boldsymbol{s}_{\boldsymbol{\theta}}, \boldsymbol{p} \rangle - \Omega(\boldsymbol{p}) \right].$$

As established earlier, the constraint $\mathbb{E}_{\boldsymbol{p}}[Y]_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]}$ strictly confines the support of $\boldsymbol{p}$ to $\mathcal{Y}^{(k)}$. We can therefore reparameterize $\boldsymbol{p}$ using the local inclusion map $\iota_{\mathcal{Y}^{(k)}} : \triangle^{\mathcal{Y}^{(k)}} \hookrightarrow \triangle^{\mathcal{Y}}$. Setting $\boldsymbol{p} = \iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q})$ for $\boldsymbol{q} \in \triangle^{\mathcal{Y}^{(k)}}$, the optimization problem restricts to the local simplex:

$$\boldsymbol{q}^* = \underset{\boldsymbol{q} \in \triangle^{\mathcal{Y}^{(k)}}}{\arg\max} \left[\langle \boldsymbol{s}_{\boldsymbol{\theta}}, \iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q}) \rangle - \Omega(\iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q})) \right].$$

By the definition of the local regularizer $\Omega^{\mathcal{Y}^{(k)}} \triangleq \Omega \circ \iota_{\mathcal{Y}^{(k)}}$, and since $\langle \boldsymbol{s}_{\boldsymbol{\theta}}, \iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q}) \rangle = \langle (\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]}, \boldsymbol{q} \rangle$, this simplifies exactly to the definition of the local smoothed maximizer:

$$\begin{aligned} \boldsymbol{q}^* &= \underset{\boldsymbol{q} \in \triangle^{\mathcal{Y}^{(k)}}}{\arg\max} \left[\langle (\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]}, \boldsymbol{q} \rangle - \Omega^{\mathcal{Y}^{(k)}}(\boldsymbol{q}) \right] \\ &= \boldsymbol{p}_{\Omega^{\mathcal{Y}^{(k)}}} \left((\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]} \right). \end{aligned}$$

Since $\boldsymbol{\mu}^{(k+1)} = \mathbb{E}_{\boldsymbol{p}^*}[Y] = \mathbb{E}_{\boldsymbol{q}^*}[Y]$, we conclude that the exact mean-space BCA update yields exactly the expected value of the distribution sampled by the RLNS algorithm. □

B.6 Proof of Proposition 6 (RLNS and Gibbs updates as Bregman projections)

First, we prove that the local RLNS update distributions match the B_Ω-projection of $\boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})$ onto the local simplex $\triangle^{\mathcal{Y}^{(k)}}$.

Proof. **Global optimization problem.** By definition, the global smoothed maximizer $\boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})$ is the unique solution to the convex optimization problem:

$$\boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}}) = \underset{\boldsymbol{p} \in \triangle^{\mathcal{Y}}}{\arg\min} \left\{ \Omega(\boldsymbol{p}) - \langle \boldsymbol{s}_{\boldsymbol{\theta}}, \boldsymbol{p} \rangle \right\}.$$

We introduce a Lagrange multiplier $\nu \in \mathbb{R}$ for the simplex equality constraint $\sum_{\boldsymbol{y} \in \mathcal{Y}} p_{\boldsymbol{y}} = 1$. The Lagrangian is $\mathcal{L}(\boldsymbol{p}, \nu) = \Omega(\boldsymbol{p}) - \langle \boldsymbol{s}_{\boldsymbol{\theta}}, \boldsymbol{p} \rangle + \nu \cdot (\langle \boldsymbol{p}, \mathbf{1} \rangle - 1)$. The first-order optimality condition gives:

$$\nabla_{\boldsymbol{p}} \mathcal{L}(\boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}}), \nu^\star) = \mathbf{0} \iff \nabla \Omega(\boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})) = \boldsymbol{s}_{\boldsymbol{\theta}} - \nu^\star \mathbf{1}. \tag{8}$$

Local optimization problem. By definition of the local regularized distribution, $\boldsymbol{p}_{\Omega^{\mathcal{Y}^{(k)}}}((\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]})$ is the solution to:

$$\begin{aligned} \boldsymbol{p}_{\Omega^{\mathcal{Y}^{(k)}}}((\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]}) &\triangleq \underset{\boldsymbol{q} \in \triangle^{\mathcal{Y}^{(k)}}}{\arg\min} \left\{ \Omega^{\mathcal{Y}^{(k)}}(\boldsymbol{q}) - \langle (\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]}, \boldsymbol{q} \rangle \right\} \\ &= \underset{\boldsymbol{q} \in \triangle^{\mathcal{Y}^{(k)}}}{\arg\min} \left\{ \Omega(\iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q})) - \langle \boldsymbol{s}_{\boldsymbol{\theta}}, \iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q}) \rangle \right\}. \end{aligned}$$

Equivalence to projection. We now analyze the projection problem. We minimize the Bregman divergence between the global solution $\boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})$ and a distribution $\boldsymbol{q} \in \triangle^{\mathcal{Y}^{(k)}}$ restricted to the local simplex:

$$B_\Omega\left(\iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q}) \| \boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})\right) = \Omega(\iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q})) - \Omega(\boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})) - \langle \nabla \Omega(\boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})), \iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q}) - \boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}}) \rangle.$$

Substituting the global optimality condition from Eq. (8) into the linear term:

$$\begin{aligned} \langle \nabla \Omega(\boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})), \iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q}) \rangle &= \langle \boldsymbol{s}_{\boldsymbol{\theta}} - \nu^\star \mathbf{1}, \iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q}) \rangle \\ &= \langle \boldsymbol{s}_{\boldsymbol{\theta}}, \iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q}) \rangle - \nu^\star \sum_{\boldsymbol{y} \in \mathcal{Y}^{(k)}} q_{\boldsymbol{y}} \\ &= \langle \boldsymbol{s}_{\boldsymbol{\theta}}, \iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q}) \rangle - \nu^\star \quad (\text{since } \boldsymbol{q} \in \triangle^{\mathcal{Y}^{(k)}}). \end{aligned}$$

Thus, we have:

$$B_\Omega\left(\iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q})\|\boldsymbol{p}_\Omega(\boldsymbol{s_\theta})\right) = \Omega(\iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q})) - \Omega(\boldsymbol{p}_\Omega(\boldsymbol{s_\theta})) - \langle \boldsymbol{s_\theta}, \iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q})\rangle + \nu^\star + \langle \nabla\Omega(\boldsymbol{p}_\Omega(\boldsymbol{s_\theta})), \boldsymbol{p}_\Omega(\boldsymbol{s_\theta})\rangle.$$

Removing terms that are constant with respect to $\boldsymbol{q}$, the projection problem reduces to:

$$\arg\min_{\boldsymbol{q}\in\triangle^{\mathcal{Y}^{(k)}}} B_\Omega\left(\iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q})\|\boldsymbol{p}_\Omega(\boldsymbol{s_\theta})\right) = \arg\min_{\boldsymbol{q}\in\triangle^{\mathcal{Y}^{(k)}}} \left\{\Omega(\iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q})) - \langle \boldsymbol{s_\theta}, \iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q})\rangle\right\}.$$

This optimization problem is identical to the one defining $\boldsymbol{p}_{\Omega^{\mathcal{Y}^{(k)}}}((\boldsymbol{s_\theta})_{[\mathcal{Y}^{(k)}]})$, proving the result. □

We now turn to the proof that the Gibbs update distributions match the KL-projection of the target onto the local simplex.

Proof. **Conditional distribution.** The conditional distribution is defined by renormalizing the global mass restricted to $\mathcal{Y}^{(k)}$:

$$\boldsymbol{p}_\Omega(\boldsymbol{s_\theta})\left(\boldsymbol{y} \mid \boldsymbol{y}^{(k)}_{[S^{(k)}]}\right) = \frac{\boldsymbol{p}_\Omega(\boldsymbol{s_\theta})(\boldsymbol{y})}{\sum_{\boldsymbol{y}'\in\mathcal{Y}^{(k)}} \boldsymbol{p}_\Omega(\boldsymbol{s_\theta})(\boldsymbol{y}')} \quad \forall \boldsymbol{y} \in \mathcal{Y}^{(k)}. \tag{9}$$

KL projection. Let $\boldsymbol{\pi} \triangleq \boldsymbol{p}_\Omega(\boldsymbol{s_\theta})$ and $\boldsymbol{q} \in \triangle^{\mathcal{Y}^{(k)}}$. We have:

$$D_{\mathrm{KL}}(\iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q})\|\boldsymbol{\pi}) = \sum_{\boldsymbol{y}\in\mathcal{Y}^{(k)}} q_{\boldsymbol{y}} \log\left(\frac{q_{\boldsymbol{y}}}{\pi_{\boldsymbol{y}}}\right).$$

We define the Lagrangian with a multiplier $\lambda \in \mathbb{R}$ for the constraint $\sum_{\boldsymbol{y}\in\mathcal{Y}^{(k)}} q_{\boldsymbol{y}} = 1$:

$$\mathcal{L}(\boldsymbol{q}, \lambda) = \sum_{\boldsymbol{y}\in\mathcal{Y}^{(k)}} q_{\boldsymbol{y}} \log(q_{\boldsymbol{y}}) - \sum_{\boldsymbol{y}\in\mathcal{Y}^{(k)}} q_{\boldsymbol{y}} \log(\pi_{\boldsymbol{y}}) + \lambda\left(\sum_{\boldsymbol{y}\in\mathcal{Y}^{(k)}} q_{\boldsymbol{y}} - 1\right).$$

Optimality conditions. The first-order condition with respect to any coordinate $q_{\boldsymbol{y}}$ is:

$$\frac{\partial\mathcal{L}}{\partial q_{\boldsymbol{y}}}(\boldsymbol{q}^\star, \lambda^\star) = 1 + \log q^\star_{\boldsymbol{y}} - \log(\pi_{\boldsymbol{y}}) + \lambda^\star = 0.$$

Solving for $q^\star_{\boldsymbol{y}}$:

$$q^\star_{\boldsymbol{y}} = \pi_{\boldsymbol{y}} \exp(-(1+\lambda^\star)).$$

Summing over $\boldsymbol{y} \in \mathcal{Y}^{(k)}$ to satisfy the simplex constraint:

$$\sum_{\boldsymbol{y}\in\mathcal{Y}^{(k)}} q^\star_{\boldsymbol{y}} = \exp(-(1+\lambda^\star)) \sum_{\boldsymbol{y}\in\mathcal{Y}^{(k)}} \pi_{\boldsymbol{y}} = 1.$$

This determines the Lagrange multiplier term: $\exp(-(1+\lambda^\star)) = (\sum_{\boldsymbol{y}'\in\mathcal{Y}^{(k)}} \pi_{\boldsymbol{y}'})^{-1}$. Substituting this back into the expression for $q^\star_{\boldsymbol{y}}$, we obtain:

$$q^\star_{\boldsymbol{y}} = \frac{\pi_{\boldsymbol{y}}}{\sum_{\boldsymbol{y}'\in\mathcal{Y}^{(k)}} \pi_{\boldsymbol{y}'}}.$$

This matches the definition of the conditional distribution in Eq. (9) exactly. □

B.7 Proof of Proposition 4 (Unbiased gradients via RLNS-1)

Proof. We first characterize the domain of $\Omega_{\boldsymbol{y}}$. We can write $F_{\boldsymbol{y}}$ as a convex combination of local maximum functions: $F_{\boldsymbol{y}}(\boldsymbol{s}) = \sum_{S\in\mathrm{supp}(\nu)} \nu(S) f_S(\boldsymbol{s})$, where $f_S(\boldsymbol{s}) \triangleq f_{\Omega^{\mathcal{Y}_S}}(\boldsymbol{s}_{[\mathcal{Y}_S]})$.

To find the domain of the conjugate f^*_S, let $h_S(\boldsymbol{u}) \triangleq f_{\Omega^{\mathcal{Y}_S}}(\boldsymbol{u})$ for $\boldsymbol{u} \in \mathbb{R}^{|\mathcal{Y}_S|}$. By definition of the smoothed maximum operator, we have $h_S = (\Omega^{\mathcal{Y}_S})^*$. Since $\Omega^{\mathcal{Y}_S}$ is strictly convex, proper, and lower

semi-continuous on the compact set $\triangle^{\mathcal{Y}_S}$, the Fenchel-Moreau theorem yields $h_S^* = (\Omega^{\mathcal{Y}_S})^{**} = \Omega^{\mathcal{Y}_S}$. Thus, $\mathrm{dom}(h_S^*) = \mathrm{dom}(\Omega^{\mathcal{Y}_S}) = \triangle^{\mathcal{Y}_S}$.

For any global score vector $\boldsymbol{s} \in \mathbb{R}^{|\mathcal{Y}|}$, we decompose it into components inside the local neighborhood, $\boldsymbol{s}_{\text{in}} \triangleq \boldsymbol{s}_{[\mathcal{Y}_S]}$, and components outside, $\boldsymbol{s}_{\text{out}} \triangleq \boldsymbol{s}_{[\mathcal{Y}\setminus\mathcal{Y}_S]}$. We can then write $f_S(\boldsymbol{s}) = h_S(\boldsymbol{s}_{\text{in}})$. By definition of the Fenchel conjugate:

$$\begin{aligned} f_S^*(\boldsymbol{p}) &= \sup_{\boldsymbol{s}_{\text{in}}, \boldsymbol{s}_{\text{out}}} \{\langle \boldsymbol{p}_{\text{in}}, \boldsymbol{s}_{\text{in}}\rangle + \langle \boldsymbol{p}_{\text{out}}, \boldsymbol{s}_{\text{out}}\rangle - h_S(\boldsymbol{s}_{\text{in}})\} \\ &= \sup_{\boldsymbol{s}_{\text{in}} \in \mathbb{R}^{|\mathcal{Y}_S|}} \{\langle \boldsymbol{p}_{\text{in}}, \boldsymbol{s}_{\text{in}}\rangle - h_S(\boldsymbol{s}_{\text{in}})\} + \sup_{\boldsymbol{s}_{\text{out}} \in \mathbb{R}^{|\mathcal{Y}\setminus\mathcal{Y}_S|}} \{\langle \boldsymbol{p}_{\text{out}}, \boldsymbol{s}_{\text{out}}\rangle\} \\ &= h_S^*(\boldsymbol{p}_{\text{in}}) + I_{\{\mathbf{0}\}}(\boldsymbol{p}_{\text{out}}), \end{aligned}$$

where $I_{\{\mathbf{0}\}}$ is 0 at $\mathbf{0}$ and $+\infty$ elsewhere. For $f_S^*(\boldsymbol{p})$ to be finite, we require $\boldsymbol{p}_{\text{out}} = \mathbf{0}$ and $\boldsymbol{p}_{\text{in}} \in \mathrm{dom}(h_S^*) = \triangle^{\mathcal{Y}_S}$. This means $\boldsymbol{p}$ must be a zero-padded distribution from the local simplex, so $\mathrm{dom}(f_S^*) = \iota_{\mathcal{Y}_S}(\triangle^{\mathcal{Y}_S})$.

We now determine the domain of $\Omega_{\boldsymbol{y}} = (F_{\boldsymbol{y}})^*$. Recall that $F_{\boldsymbol{y}}$ is a sum of scaled functions:

$$F_{\boldsymbol{y}}(\boldsymbol{s}) = \sum_{S \in \mathrm{supp}(\nu)} \nu(S) f_S(\boldsymbol{s}).$$

Let $g_S(\boldsymbol{s}) \triangleq \nu(S) f_S(\boldsymbol{s})$. A known property of the Fenchel conjugate then gives that for any $\nu(S) > 0$, we have:

$$g_S^*(\boldsymbol{p}) = \nu(S) f_S^*\left(\frac{\boldsymbol{p}}{\nu(S)}\right).$$

Consequently, the effective domain of the scaled conjugate is scaled by the same factor:

$$\boldsymbol{p} \in \mathrm{dom}(g_S^*) \iff \frac{\boldsymbol{p}}{\nu(S)} \in \mathrm{dom}(f_S^*) \iff \boldsymbol{p} \in \nu(S)\mathrm{dom}(f_S^*).$$

Thus, $\mathrm{dom}(g_S^*) = \nu(S)\mathrm{dom}(f_S^*)$.

The Fenchel conjugate of a sum of functions is the inf-convolution (denoted by $\square$) of their individual conjugates [Rockafellar, 1970]:

$$\Omega_{\boldsymbol{y}} = (F_{\boldsymbol{y}})^* = \square_{S \in \mathrm{supp}(\nu)} g_S^*.$$

Since the effective domain of the inf-convolution is the Minkowski sum of the effective domains, we then have:

$$\begin{aligned} \mathrm{dom}(\Omega_{\boldsymbol{y}}) &= \sum_{S \in \mathrm{supp}(\nu)} \mathrm{dom}(g_S^*) \\ &= \sum_{S \in \mathrm{supp}(\nu)} \nu(S)\mathrm{dom}(f_S^*) \\ &= \sum_{S \in \mathrm{supp}(\nu)} \nu(S)\iota_{\mathcal{Y}_S}(\triangle^{\mathcal{Y}_S}). \end{aligned}$$

Because $\boldsymbol{y}$ is the reference solution defining the local neighborhoods, it satisfies $\boldsymbol{y} \in \mathcal{Y}_S$ for every subset S. Thus, we have $\delta_{\boldsymbol{y}}^{\mathcal{Y}_S} \in \triangle^{\mathcal{Y}_S}$, and by padding, we get $\delta_{\boldsymbol{y}}^{\mathcal{Y}} = \iota_{\mathcal{Y}_S}(\delta_{\boldsymbol{y}}^{\mathcal{Y}_S}) \in \iota_{\mathcal{Y}_S}(\triangle^{\mathcal{Y}_S})$. Picking this specific point from each set in the Minkowski sum yields:

$$\sum_{S \in \mathrm{supp}(\nu)} \nu(S)\delta_{\boldsymbol{y}}^{\mathcal{Y}} = \left(\sum_{S \in \mathrm{supp}(\nu)} \nu(S)\right) \cdot \delta_{\boldsymbol{y}}^{\mathcal{Y}} = \delta_{\boldsymbol{y}}^{\mathcal{Y}}.$$

This proves that $\delta_{\boldsymbol{y}}^{\mathcal{Y}} \in \mathrm{dom}(\Omega_{\boldsymbol{y}})$, ensuring the target-dependent Fenchel-Young loss is well-defined and finite at the ground-truth distribution.

Now, assuming that ν does not depend on $\boldsymbol{s}$, we can exchange the expectation and the gradient signs. The gradient of $F_{\boldsymbol{y}}$ is then given by Danskin's theorem and the chain rule:

$$\nabla_{\boldsymbol{s}} F_{\boldsymbol{y}}(\boldsymbol{s}) = \mathbb{E}_{S\sim\nu}\left[\iota_{\mathcal{Y}_S}\left(\boldsymbol{p}_{\Omega^{\mathcal{Y}_S}}\left(\boldsymbol{s}_{[\mathcal{Y}_S]}\right)\right)\right].$$

The gradient of the target-dependent loss $L_{\Omega_{\boldsymbol{y}}}$ with respect to $\boldsymbol{\theta}$ at $\boldsymbol{p} = \delta^{\mathcal{Y}}_{\boldsymbol{y}}$ is:

$$\nabla_{\boldsymbol{\theta}} L_{\Omega_{\boldsymbol{y}}}(\boldsymbol{\theta}\,;\delta^{\mathcal{Y}}_{\boldsymbol{y}}) = \nabla_{\boldsymbol{\theta}} F_{\boldsymbol{y}}(\boldsymbol{s}_{\boldsymbol{\theta}}) - \boldsymbol{y} = \mathbb{E}_{S\sim\nu}\left[\mathbb{E}_{Y\sim\boldsymbol{p}_{\Omega^{\mathcal{Y}_S}}\left((\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}_S]}\right)}[Y]\right] - \boldsymbol{y}.$$

This exactly matches the expected update of RLNS:

$$S\sim\nu \quad\text{and}\quad Y \mid S \sim \boldsymbol{p}_{\Omega^{\mathcal{Y}_S}}\left((\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}_S]}\right) \implies \mathbb{E}_S[\mathbb{E}[Y\mid S]] - \boldsymbol{y} = \nabla_{\boldsymbol{\theta}} L_{\Omega_{\boldsymbol{y}}}(\boldsymbol{\theta}\,;\delta^{\mathcal{Y}}_{\boldsymbol{y}}).$$

Thus, RLNS sampling initialized at $\boldsymbol{y}$ and run for 1 step produces an unbiased stochastic gradient. □

B.8 Proof of Proposition 5 (Equivalence to pseudolikelihood)

Proof. In the Shannon entropy-regularized setting $\Omega = -\gamma H^s$, the smoothed local maximum evaluates to the log-sum-exp function:

$$\begin{aligned} f_{\Omega^{\mathcal{Y}_S}}\left(\boldsymbol{s}_{[\mathcal{Y}_S]}\right) &= \max_{\boldsymbol{q}\in\triangle^{\mathcal{Y}_S}} \langle \boldsymbol{q}, \boldsymbol{s}_{[\mathcal{Y}_S]}\rangle - \Omega\circ\iota_{\mathcal{Y}_S}(\boldsymbol{q}) \\ &= \max_{\boldsymbol{q}\in\triangle^{\mathcal{Y}_S}} \langle \boldsymbol{q}, \boldsymbol{s}_{[\mathcal{Y}_S]}\rangle + \gamma H^s_{\mathcal{Y}_S}(\boldsymbol{q}) \quad \text{where } H^s_{\mathcal{Y}_S}(\boldsymbol{q}) \triangleq -\sum_{\boldsymbol{y}'\in\mathcal{Y}_S} q_{\boldsymbol{y}'}\log(q_{\boldsymbol{y}'}) \\ &= \gamma\log\sum_{\boldsymbol{y}'\in\mathcal{Y}_S}\exp\left(s_{\boldsymbol{y}'}/\gamma\right). \end{aligned}$$

We first determine the value of the additive constant $\Omega_{\boldsymbol{y}}(\delta^{\mathcal{Y}}_{\boldsymbol{y}})$. By definition of the convex conjugate, we have:

$$\Omega_{\boldsymbol{y}}(\delta^{\mathcal{Y}}_{\boldsymbol{y}}) = \sup_{\boldsymbol{s}\in\mathbb{R}^{|\mathcal{Y}|}}\left\{\langle \boldsymbol{s}, \delta^{\mathcal{Y}}_{\boldsymbol{y}}\rangle - F_{\boldsymbol{y}}(\boldsymbol{s})\right\} = \sup_{\boldsymbol{s}\in\mathbb{R}^{|\mathcal{Y}|}}\left\{s_{\boldsymbol{y}} - \mathbb{E}_{S\sim\nu}\left[\gamma\log\sum_{\boldsymbol{y}'\in\mathcal{Y}_S}\exp(s_{\boldsymbol{y}'}/\gamma)\right]\right\}.$$

Notice that for any subset S, the ground-truth solution $\boldsymbol{y}$ always belongs to its own local neighborhood, meaning $\boldsymbol{y}\in\mathcal{Y}_S$. Because all terms in the exponential sum are strictly positive, the sum is lower-bounded by the single term corresponding to $\boldsymbol{y}$:

$$\sum_{\boldsymbol{y}'\in\mathcal{Y}_S}\exp(s_{\boldsymbol{y}'}/\gamma) \geq \exp(s_{\boldsymbol{y}}/\gamma).$$

Taking the logarithm and multiplying by $\gamma > 0$, we get:

$$\gamma\log\sum_{\boldsymbol{y}'\in\mathcal{Y}_S}\exp(s_{\boldsymbol{y}'}/\gamma) \geq \gamma\log(\exp(s_{\boldsymbol{y}}/\gamma)) = s_{\boldsymbol{y}}.$$

Therefore, for any score vector $\boldsymbol{s}\in\mathbb{R}^{|\mathcal{Y}|}$ and any subset S, the difference inside the expectation is non-positive:

$$s_{\boldsymbol{y}} - \gamma\log\sum_{\boldsymbol{y}'\in\mathcal{Y}_S}\exp(s_{\boldsymbol{y}'}/\gamma) \leq 0.$$

Taking the expectation over $S\sim\nu$ preserves this inequality, meaning $\Omega_{\boldsymbol{y}}(\delta^{\mathcal{Y}}_{\boldsymbol{y}}) \leq 0$. To show that the supremum is exactly 0, we can construct a sequence of score vectors that asymptotically achieves it. Let $s_{\boldsymbol{y}} = 0$, and for all $\boldsymbol{y}'\neq\boldsymbol{y}$, let $s_{\boldsymbol{y}'}\to-\infty$. Under this limit, for any subset S:

$$\lim_{s_{\boldsymbol{y}'\neq\boldsymbol{y}}\to-\infty}\gamma\log\sum_{\boldsymbol{y}'\in\mathcal{Y}_S}\exp(s_{\boldsymbol{y}'}/\gamma) = \gamma\log(\exp(0)+0) = 0.$$

Consequently, the objective value tends to $0 - 0 = 0$. Since the supremum is bounded above by 0 and can be asymptotically approached, we conclude that $\Omega_{\boldsymbol{y}}(\delta_{\boldsymbol{y}}^{\mathcal{Y}}) = 0$.

Now, substituting $\boldsymbol{s}_{\boldsymbol{\theta}} = (\langle \boldsymbol{\theta}, \boldsymbol{y}' \rangle)_{\boldsymbol{y}' \in \mathcal{Y}}$ and $\Omega_{\boldsymbol{y}}(\delta_{\boldsymbol{y}}^{\mathcal{Y}}) = 0$ into the loss function:

$$\begin{aligned} L_{\Omega_{\boldsymbol{y}}}(\boldsymbol{\theta}\,; \delta_{\boldsymbol{y}}^{\mathcal{Y}}) &= F_{\boldsymbol{y}}(\boldsymbol{s}_{\boldsymbol{\theta}}) + \Omega_{\boldsymbol{y}}(\delta_{\boldsymbol{y}}^{\mathcal{Y}}) - \langle \boldsymbol{\theta}, \boldsymbol{y} \rangle \\ &= \mathbb{E}_{S \sim \nu} \left[f_{\Omega^{\mathcal{Y}_S}} \left((\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}_S]} \right) \right] + 0 - \langle \boldsymbol{\theta}, \boldsymbol{y} \rangle \\ &= \mathbb{E}_{S \sim \nu} \left[\gamma \log \sum_{\boldsymbol{y}' \in \mathcal{Y}_S} \exp(\langle \boldsymbol{\theta}, \boldsymbol{y}' \rangle / \gamma) - \langle \boldsymbol{\theta}, \boldsymbol{y} \rangle \right]. \end{aligned}$$

Moreover, denoting $\pi_{\boldsymbol{\theta}} \triangleq \boldsymbol{p}_{\Omega}(\boldsymbol{s}_{\boldsymbol{\theta}})$, we know that $\pi_{\boldsymbol{\theta}}(\boldsymbol{y}) \propto \exp(\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle / \gamma)$ in this setting. Thus, the conditional probability is:

$$\begin{aligned} \pi_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{y}_{[S]}) &= \frac{\pi_{\boldsymbol{\theta}}(\boldsymbol{y})}{\sum_{\boldsymbol{y}' \in \mathcal{Y}_S} \pi_{\boldsymbol{\theta}}(\boldsymbol{y}')} \\ &= \frac{\exp(\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle / \gamma)}{\sum_{\boldsymbol{y}'' \in \mathcal{Y}} \exp(\langle \boldsymbol{\theta}, \boldsymbol{y}'' \rangle / \gamma)} \cdot \left(\frac{\sum_{\boldsymbol{y}' \in \mathcal{Y}_S} \exp(\langle \boldsymbol{\theta}, \boldsymbol{y}' \rangle / \gamma)}{\sum_{\boldsymbol{y}'' \in \mathcal{Y}} \exp(\langle \boldsymbol{\theta}, \boldsymbol{y}'' \rangle / \gamma)} \right)^{-1} \\ &= \frac{\exp(\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle / \gamma)}{\sum_{\boldsymbol{y}' \in \mathcal{Y}_S} \exp(\langle \boldsymbol{\theta}, \boldsymbol{y}' \rangle / \gamma)}. \end{aligned}$$

Taking the negative logarithm and multiplying by γ:

$$-\gamma \log \pi_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{y}_{[S]}) = \gamma \log \sum_{\boldsymbol{y}' \in \mathcal{Y}_S} \exp(\langle \boldsymbol{\theta}, \boldsymbol{y}' \rangle / \gamma) - \langle \boldsymbol{\theta}, \boldsymbol{y} \rangle.$$

Substituting this back into the loss expression gives the final result:

$$L_{\Omega_{\boldsymbol{y}}}(\boldsymbol{\theta}\,; \delta_{\boldsymbol{y}}^{\mathcal{Y}}) = \mathbb{E}_{S \sim \nu} \left[-\gamma \log \pi_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{y}_{[S]}) \right].$$

□

B.9 Proof of Proposition 8 (Bias of RLNS)

Proof. This result follows from the perturbation bounds for uniformly ergodic Markov chains established by Mitrophanov [2005]. Specifically, we apply Theorem 3.2 and Corollary 3.2 from that work.

Let π^* be the stationary distribution of P^*. Mitrophanov [2005] shows that if $\tau_m(P^*) < 1$, then:

$$d_{\mathrm{TV}}\left(\pi_{\boldsymbol{\theta},\Omega}^{\mathrm{LNS}}, \pi^*\right) \leq \frac{\max_{\boldsymbol{y} \in \mathcal{Y}} d_{\mathrm{TV}}\left(P^m(\boldsymbol{y}, \cdot), (P^*)^m(\boldsymbol{y}, \cdot)\right)}{1 - \tau_m(P^*)}$$

(note that the original theorem is stated in terms of operator norms, which corresponds to exactly twice the TV distance for both the numerator and the left-hand side: the factors of 2 cancel out).

Furthermore, as shown in the proof of Corollary 3.2 in Mitrophanov [2005], the error over m steps is bounded by m times the single-step error:

$$\max_{\boldsymbol{y} \in \mathcal{Y}} d_{\mathrm{TV}}\left(P^m(\boldsymbol{y}, \cdot), (P^*)^m(\boldsymbol{y}, \cdot)\right) \leq m \cdot \max_{\boldsymbol{y} \in \mathcal{Y}} d_{\mathrm{TV}}\left(P(\boldsymbol{y}, \cdot), P^*(\boldsymbol{y}, \cdot)\right).$$

We now bound the distance between the transition kernels. Let ν denote the probability distribution used to select the subset of coordinates $S^{(k)}$ in Step 3 of Algorithm 2 (and identically for the ideal Gibbs sampler). For a fixed current state $\boldsymbol{y}^{(k)} \in \mathcal{Y}$, the transition kernel $P(\boldsymbol{y}^{(k)}, \cdot)$ is a mixture of the local update distributions over all possible choices of $S^{(k)}$. Using the inclusion map $\iota_{\mathcal{Y}^{(k)}}$ from Definition 2 to map local distributions to $\triangle^{\mathcal{Y}}$, we denote:

$$\begin{aligned} \boldsymbol{q}^{\mathrm{LNS}}(\boldsymbol{y}^{(k)}, S^{(k)}) &\triangleq \iota_{\mathcal{Y}^{(k)}}\left(\boldsymbol{p}_{\Omega^{\mathcal{Y}^{(k)}}}\left((\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]}\right)\right) \in \triangle^{\mathcal{Y}}, \\ \boldsymbol{q}^{\mathrm{Gibbs}}(\boldsymbol{y}^{(k)}, S^{(k)}) &\triangleq \iota_{\mathcal{Y}^{(k)}}\left(\boldsymbol{p}_{\Omega}\left(\boldsymbol{s}_{\boldsymbol{\theta}}\right)\left(\cdot \mid \boldsymbol{y}^{(k)}_{[S^{(k)}]}\right)\right) \in \triangle^{\mathcal{Y}}. \end{aligned}$$

Then, we have:

$$P(\boldsymbol{y}^{(k)}, \cdot) = \sum_{S^{(k)} \subseteq [d]} \nu(S^{(k)})\, \boldsymbol{q}^{\mathrm{LNS}}(\boldsymbol{y}^{(k)}, S^{(k)}).$$

Similarly, the ideal Gibbs kernel $P^*(\boldsymbol{y}^{(k)}, \cdot)$ uses the conditional distributions:

$$P^*(\boldsymbol{y}^{(k)}, \cdot) = \sum_{S^{(k)} \subseteq [d]} \nu(S^{(k)})\, \boldsymbol{q}^{\mathrm{Gibbs}}(\boldsymbol{y}^{(k)}, S^{(k)}).$$

We compute the total variation distance between these two mixtures. Using the convexity of the TV distance, we have:

$$\begin{aligned} d_{\mathrm{TV}}\big(P(\boldsymbol{y}^{(k)}, \cdot), P^*(\boldsymbol{y}^{(k)}, \cdot)\big) &= d_{\mathrm{TV}}\Big(\sum_{S^{(k)}} \nu(S^{(k)}) \boldsymbol{q}^{\mathrm{LNS}}(\boldsymbol{y}^{(k)}, S^{(k)}), \sum_{S^{(k)}} \nu(S^{(k)}) \boldsymbol{q}^{\mathrm{Gibbs}}(\boldsymbol{y}^{(k)}, S^{(k)})\Big) \\ &\leq \sum_{S^{(k)} \subseteq [d]} \nu(S^{(k)})\, d_{\mathrm{TV}}\big(\boldsymbol{q}^{\mathrm{LNS}}(\boldsymbol{y}^{(k)}, S^{(k)}), \boldsymbol{q}^{\mathrm{Gibbs}}(\boldsymbol{y}^{(k)}, S^{(k)})\big). \end{aligned}$$

Since we have $d_{\mathrm{TV}}(\boldsymbol{p}, \boldsymbol{p}') = d_{\mathrm{TV}}(\iota_{\mathcal{Y}^{(k)}}(\boldsymbol{p}), \iota_{\mathcal{Y}^{(k)}}(\boldsymbol{p}'))$ for any $\boldsymbol{p}, \boldsymbol{p}' \in \triangle^{\mathcal{Y}^{(k)}}$, Definition 6 then gives that this local distance is bounded by $\delta_{\boldsymbol{\theta},\Omega}$ for every $\boldsymbol{y}^{(k)}$ and $S^{(k)}$:

$$d_{\mathrm{TV}}\big(\boldsymbol{q}^{\mathrm{LNS}}(\boldsymbol{y}^{(k)}, S^{(k)}), \boldsymbol{q}^{\mathrm{Gibbs}}(\boldsymbol{y}^{(k)}, S^{(k)})\big) \leq \max_{\boldsymbol{y}', S'} d_{\mathrm{TV}}\big(\boldsymbol{q}^{\mathrm{LNS}}(\boldsymbol{y}', S'), \boldsymbol{q}^{\mathrm{Gibbs}}(\boldsymbol{y}', S')\big) = \delta_{\boldsymbol{\theta},\Omega}.$$

Substituting this into the sum:

$$\begin{aligned} d_{\mathrm{TV}}\left(P(\boldsymbol{y}^{(k)}, \cdot), P^*(\boldsymbol{y}^{(k)}, \cdot)\right) &\leq \sum_{S^{(k)} \subseteq [d]} \nu(S^{(k)})\, \delta_{\boldsymbol{\theta},\Omega} \\ &= \delta_{\boldsymbol{\theta},\Omega}. \end{aligned}$$

Taking the maximum over $\boldsymbol{y}^{(k)} \in \mathcal{Y}$ completes the proof. □

B.10 Proof of Proposition 9 (Consistency gap bound via relative smoothness)

Proof. Let $\boldsymbol{\pi} \triangleq \boldsymbol{p}_\Omega(\boldsymbol{s}_{\boldsymbol{\theta}})$ be the global regularized distribution. Fix a neighborhood $\mathcal{Y}^{(k)}$ defined by some $\boldsymbol{y}^{(k)} \in \mathcal{Y}$, and $S^{(k)} \subseteq [d]$. Let $\hat{\boldsymbol{q}}, \boldsymbol{q}^* \in \triangle^{\mathcal{Y}^{(k)}}$ denote the RLNS update and the ideal Gibbs update distributions (the projections of $\boldsymbol{\pi}$ onto $\triangle^{\mathcal{Y}^{(k)}}$ via B_Ω and $B_{-\gamma H^s}$, per Proposition 6).

Optimality conditions of the projection problems. Both $\hat{\boldsymbol{q}}$ and $\boldsymbol{q}^*$ satisfy the first-order optimality conditions for their respective projection problems. For any $\boldsymbol{z} \in \triangle^{\mathcal{Y}^{(k)}}$:

$$\begin{aligned} \langle \nabla_{\hat{\boldsymbol{q}}} B_\Omega\left(\iota_{\mathcal{Y}^{(k)}}(\hat{\boldsymbol{q}}) \| \boldsymbol{\pi}\right), \boldsymbol{z} - \hat{\boldsymbol{q}} \rangle \geq 0, \\ \langle \nabla_{\boldsymbol{q}^*} B_{-\gamma H^s}\left(\iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q}^*) \| \boldsymbol{\pi}\right), \boldsymbol{z} - \boldsymbol{q}^* \rangle \geq 0. \end{aligned}$$

Since for any f we have $\nabla_1 B_f(\boldsymbol{u} \| \boldsymbol{v}) = \nabla f(\boldsymbol{u}) - \nabla f(\boldsymbol{v})$, applying the chain rule gives:

$$\begin{aligned} \left\langle [\nabla \iota_{\mathcal{Y}^{(k)}}(\hat{\boldsymbol{q}})]^\top \left(\nabla\Omega(\iota_{\mathcal{Y}^{(k)}}(\hat{\boldsymbol{q}})) - \nabla\Omega(\boldsymbol{\pi})\right), \boldsymbol{z} - \hat{\boldsymbol{q}} \right\rangle \geq 0, \\ \left\langle [\nabla \iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q}^*)]^\top \left(\nabla(-\gamma H^s)(\iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q}^*)) - \nabla(-\gamma H^s)(\boldsymbol{\pi})\right), \boldsymbol{z} - \boldsymbol{q}^* \right\rangle \geq 0. \end{aligned}$$

Since $\iota_{\mathcal{Y}^{(k)}}$ is a linear map, it is its own Jacobian. Using the identity $\langle \boldsymbol{A}^\top \boldsymbol{x}, \boldsymbol{y} \rangle = \langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{y} \rangle$, we get:

$$\langle \nabla\Omega(\iota_{\mathcal{Y}^{(k)}}(\hat{\boldsymbol{q}})) - \nabla\Omega(\boldsymbol{\pi}), \iota_{\mathcal{Y}^{(k)}}(\boldsymbol{z} - \hat{\boldsymbol{q}}) \rangle \geq 0, \tag{10}$$

$$\langle \nabla(-\gamma H^s)(\iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q}^*)) - \nabla(-\gamma H^s)(\boldsymbol{\pi}), \iota_{\mathcal{Y}^{(k)}}(\boldsymbol{z} - \boldsymbol{q}^*) \rangle \geq 0. \tag{11}$$

Decomposition. We now decompose the gradient of Ω as $\nabla\Omega = \nabla(-\gamma H^s) + \nabla h_\gamma$. Substituting this into Eq. (10) and applying it to $\boldsymbol{z} = \boldsymbol{q}^* \in \triangle^{\mathcal{Y}^{(k)}}$, we get:

$$\langle \nabla(-\gamma H^s)(\iota_{\mathcal{Y}^{(k)}}(\hat{\boldsymbol{q}})) + \nabla h_\gamma(\iota_{\mathcal{Y}^{(k)}}(\hat{\boldsymbol{q}})) - \nabla(-\gamma H^s)(\boldsymbol{\pi}) - \nabla h_\gamma(\boldsymbol{\pi}), \iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q}^* - \hat{\boldsymbol{q}}) \rangle \geq 0. \tag{12}$$

Next, we apply Eq. (11) to $\boldsymbol{z} = \hat{\boldsymbol{q}} \in \triangle^{\mathcal{Y}^{(k)}}$:

$$\langle \nabla(-\gamma H^s)(\iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q}^*)) - \nabla(-\gamma H^s)(\boldsymbol{\pi}), \iota_{\mathcal{Y}^{(k)}}(\hat{\boldsymbol{q}} - \boldsymbol{q}^*)\rangle \geq 0. \tag{13}$$

Summing Eqs. (12) and (13) and recognizing that $\iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q}^* - \hat{\boldsymbol{q}}) = -\iota_{\mathcal{Y}^{(k)}}(\hat{\boldsymbol{q}} - \boldsymbol{q}^*)$, the $\nabla(-\gamma H^s)(\boldsymbol{\pi})$ terms cancel out, and we obtain:

$$\begin{aligned}\langle \nabla(-\gamma H^s)(\iota_{\mathcal{Y}^{(k)}}(\hat{\boldsymbol{q}})) - \nabla(-\gamma H^s)(\iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q}^*)), \iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q}^* - \hat{\boldsymbol{q}})\rangle \\ + \langle \nabla h_\gamma(\iota_{\mathcal{Y}^{(k)}}(\hat{\boldsymbol{q}})) - \nabla h_\gamma(\boldsymbol{\pi}), \iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q}^* - \hat{\boldsymbol{q}})\rangle \geq 0.\end{aligned}$$

Rearranging the terms and using the linearity of $\iota_{\mathcal{Y}^{(k)}}$ yields:

$$\begin{aligned}&\langle \nabla h_\gamma(\iota_{\mathcal{Y}^{(k)}}(\hat{\boldsymbol{q}})) - \nabla h_\gamma(\boldsymbol{\pi}), \iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q}^* - \hat{\boldsymbol{q}})\rangle \\ &\geq \langle \nabla(-\gamma H^s)(\iota_{\mathcal{Y}^{(k)}}(\hat{\boldsymbol{q}})) - \nabla(-\gamma H^s)(\iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q}^*)), \iota_{\mathcal{Y}^{(k)}}(\hat{\boldsymbol{q}}) - \iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q}^*)\rangle\,.\end{aligned} \tag{14}$$

Strong convexity of $-\gamma H^s$. By Pinsker's inequality, $-H^s$ is 1-strongly convex with respect to the ℓ_1 norm, which implies that $-\gamma H^s$ is γ-strongly convex with respect to the ℓ_1 norm. Thus, for any $\boldsymbol{u}, \boldsymbol{v} \in \triangle^{\mathcal{Y}}$, we have:

$$\langle \nabla(-\gamma H^s)(\boldsymbol{u}) - \nabla(-\gamma H^s)(\boldsymbol{v})\,,\, \boldsymbol{u} - \boldsymbol{v}\rangle \geq \gamma \|\boldsymbol{u} - \boldsymbol{v}\|_1^2. \tag{15}$$

The right-hand side of Eq. (14) is now bounded by the strong convexity of $-\gamma H^s$, and Eq. (15) gives:

$$\begin{aligned}\langle \nabla(-\gamma H^s)(\iota_{\mathcal{Y}^{(k)}}(\hat{\boldsymbol{q}})) - \nabla(-\gamma H^s)(\iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q}^*)), \iota_{\mathcal{Y}^{(k)}}(\hat{\boldsymbol{q}}) - \iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q}^*)\rangle &\geq \gamma \|\iota_{\mathcal{Y}^{(k)}}(\hat{\boldsymbol{q}}) - \iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q}^*)\|_1^2 \\ &\geq \gamma \|\hat{\boldsymbol{q}} - \boldsymbol{q}^*\|_1^2.\end{aligned}$$

Combining this with Eq. (14), we get:

$$\gamma \|\hat{\boldsymbol{q}} - \boldsymbol{q}^*\|_1^2 \leq \langle \nabla h_\gamma(\iota_{\mathcal{Y}^{(k)}}(\hat{\boldsymbol{q}})) - \nabla h_\gamma(\boldsymbol{\pi}), \iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q}^* - \hat{\boldsymbol{q}})\rangle. \tag{16}$$

Smoothness of h_γ. Applying Hölder's inequality to the right-hand side of Eq. (16):

$$\begin{aligned}\gamma \|\hat{\boldsymbol{q}} - \boldsymbol{q}^*\|_1^2 &\leq \|\nabla h_\gamma(\iota_{\mathcal{Y}^{(k)}}(\hat{\boldsymbol{q}})) - \nabla h_\gamma(\boldsymbol{\pi})\|_\infty \cdot \|\iota_{\mathcal{Y}^{(k)}}(\boldsymbol{q}^* - \hat{\boldsymbol{q}})\|_1 \\ &\leq \|\nabla h_\gamma(\iota_{\mathcal{Y}^{(k)}}(\hat{\boldsymbol{q}})) - \nabla h_\gamma(\boldsymbol{\pi})\|_\infty \cdot \|\hat{\boldsymbol{q}} - \boldsymbol{q}^*\|_1.\end{aligned}$$

Assuming $\hat{\boldsymbol{q}} \neq \boldsymbol{q}^*$, dividing by $\|\hat{\boldsymbol{q}} - \boldsymbol{q}^*\|_1$, and using the definition $d_{\mathrm{TV}}(\boldsymbol{p}, \boldsymbol{q}) = \frac{1}{2}\|\boldsymbol{p} - \boldsymbol{q}\|_1$ yields:

$$d_{\mathrm{TV}}(\hat{\boldsymbol{q}}\,, \boldsymbol{q}^*) \leq \frac{1}{2\gamma} \|\nabla h_\gamma(\iota_{\mathcal{Y}^{(k)}}(\hat{\boldsymbol{q}})) - \nabla h_\gamma(\boldsymbol{\pi})\|_\infty.$$

Then, from the $L_{\Omega,\gamma}$-smoothness of h_γ for the ℓ_1 norm (whose dual is the ℓ_∞ norm), we get:

$$d_{\mathrm{TV}}(\hat{\boldsymbol{q}}\,, \boldsymbol{q}^*) \leq \frac{L_{\Omega,\gamma}}{2\gamma} \|\iota_{\mathcal{Y}^{(k)}}(\hat{\boldsymbol{q}}) - \boldsymbol{\pi}\|_1.$$

Since $\boldsymbol{\pi}, \iota_{\mathcal{Y}^{(k)}}(\hat{\boldsymbol{q}}) \in \triangle^{\mathcal{Y}}$, we conclude by using the fact that the diameter of $\triangle^{\mathcal{Y}}$ in ℓ_1 norm is 2, which finally gives:

$$d_{\mathrm{TV}}(\hat{\boldsymbol{q}}\,, \boldsymbol{q}^*) \leq \frac{L_{\Omega,\gamma}}{\gamma}.$$

Since this bound is independent of the chosen $\boldsymbol{y}^{(k)}$ and $S^{(k)}$ (which determine $\hat{\boldsymbol{q}}$ and $\boldsymbol{q}^*$ by determining $\mathcal{Y}^{(k)}$), we finally get:

$$\delta_{\boldsymbol{\theta},\Omega} \triangleq \max_{\boldsymbol{y}^{(k)}, S^{(k)}} d_{\mathrm{TV}}(\hat{\boldsymbol{q}}, \boldsymbol{q}^*) \leq \frac{L_{\Omega,\gamma}}{\gamma}.$$

□

B.11 Proof of Proposition 10 (Consistency gap bound in the perturbation-based setting)

Proof. Consider a fixed current solution $\boldsymbol{y}^{(k)} \in \mathcal{Y}$ and a subset of coordinates $S^{(k)} \subseteq [d]$, defining the local neighborhood $\mathcal{Y}^{(k)}$.

Let $\hat{\boldsymbol{q}} \triangleq \boldsymbol{p}_{\Omega^{\mathcal{Y}^{(k)}}_{\varepsilon}}((\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]})$ denote the local regularized distribution, $\pi^* \triangleq \boldsymbol{p}_{\Omega_\varepsilon}(\boldsymbol{s}_{\boldsymbol{\theta}})$ denote the global regularized distribution, and $\boldsymbol{q}^* \triangleq \boldsymbol{p}_{\Omega_\varepsilon}(\boldsymbol{s}_{\boldsymbol{\theta}})(\cdot|\boldsymbol{y}^{(k)}_{[S^{(k)}]})$ denote the global regularized distribution conditioned on $\mathcal{Y}^{(k)}$. We rely on the coupling induced by the shared noise variable Z.

Let $Z \sim \mathcal{D}$ be a random noise vector. We define two coupled random variables taking values in $\mathcal{Y}$ and $\mathcal{Y}^{(k)}$ respectively:

$$Y^* \triangleq \arg\max_{\boldsymbol{y}\in\mathcal{Y}}\{\langle \boldsymbol{\theta} + \varepsilon Z, \boldsymbol{y}\rangle\},$$
$$\hat{Y} \triangleq \arg\max_{\boldsymbol{y}\in\mathcal{Y}^{(k)}}\{\langle \boldsymbol{\theta} + \varepsilon Z, \boldsymbol{y}\rangle\}.$$

By definition of the perturbation-based regularizer Ω_ε (Definition 4) and Proposition 3, the distributions of these variables correspond exactly to the global and local regularized distributions:

$$Y^* \sim \pi^* \quad \text{and} \quad \hat{Y} \sim \hat{\boldsymbol{q}}.$$

Coupling event. Let E be the event that the global perturbed maximizer falls within the local neighborhood:

$$E \triangleq \{Y^* \in \mathcal{Y}^{(k)}\}.$$

Since $\mathcal{Y}^{(k)} \subseteq \mathcal{Y}$, if the maximum over the larger set $\mathcal{Y}$ belongs to the subset $\mathcal{Y}^{(k)}$, it must strictly coincide with the maximum over the subset $\mathcal{Y}^{(k)}$ (assuming unique maximizers almost surely, which holds if $\mathcal{D}$ has a density). Therefore:

$$\forall \omega \in E, \quad Y^*(\omega) = \hat{Y}(\omega). \tag{17}$$

We denote the probability of this event by $\alpha \triangleq \mathbb{P}(E) = \pi^*(\mathcal{Y}^{(k)})$.

Decomposition of the local distribution. The ideal conditional distribution $\boldsymbol{q}^*$ corresponds to the distribution of Y^* conditioned on E:

$$\forall \boldsymbol{y} \in \mathcal{Y}^{(k)}, \quad q^*_{\boldsymbol{y}} = \mathbb{P}(Y^* = \boldsymbol{y} \mid E).$$

We decompose the probability mass function of $\hat{Y}$ (which is $\hat{\boldsymbol{q}}$) by conditioning on E:

$$\hat{q}_{\boldsymbol{y}} = \mathbb{P}(\hat{Y} = \boldsymbol{y}) = \mathbb{P}(\hat{Y} = \boldsymbol{y} \mid E)\mathbb{P}(E) + \mathbb{P}(\hat{Y} = \boldsymbol{y} \mid \overline{E})\mathbb{P}(\overline{E}).$$

Using the coupling implication from Eq. (17), we have $\hat{Y} = Y^*$ on E. Thus:

$$\mathbb{P}(\hat{Y} = \boldsymbol{y} \mid E) = \mathbb{P}(Y^* = \boldsymbol{y} \mid E) = q^*_{\boldsymbol{y}}.$$

Substituting this back, and letting $\boldsymbol{r} \in \triangle^{\mathcal{Y}^{(k)}}$ be defined by $r_{\boldsymbol{y}} \triangleq \mathbb{P}(\hat{Y} = \boldsymbol{y} \mid \overline{E})$, we obtain the mixture:

$$\hat{q}_{\boldsymbol{y}} = \alpha q^*_{\boldsymbol{y}} + (1-\alpha) r_{\boldsymbol{y}}.$$

Total variation bound. We now compute the total variation distance between $\hat{\boldsymbol{q}}$ and $\boldsymbol{q}^*$:

$$\begin{aligned}
d_{\mathrm{TV}}(\hat{\boldsymbol{q}}, \boldsymbol{q}^*) &= \frac{1}{2} \sum_{\boldsymbol{y}\in\mathcal{Y}^{(k)}} |\hat{q}_{\boldsymbol{y}} - q^*_{\boldsymbol{y}}| \\
&= \frac{1}{2} \sum_{\boldsymbol{y}\in\mathcal{Y}^{(k)}} |\alpha q^*_{\boldsymbol{y}} + (1-\alpha) r_{\boldsymbol{y}} - q^*_{\boldsymbol{y}}| \\
&= \frac{1}{2} \sum_{\boldsymbol{y}\in\mathcal{Y}^{(k)}} |(1-\alpha)(r_{\boldsymbol{y}} - q^*_{\boldsymbol{y}})| \\
&= (1-\alpha) \left(\frac{1}{2} \sum_{\boldsymbol{y}\in\mathcal{Y}^{(k)}} |r_{\boldsymbol{y}} - q^*_{\boldsymbol{y}}| \right) \\
&= (1-\alpha) \cdot d_{\mathrm{TV}}(\boldsymbol{r}, \boldsymbol{q}^*).
\end{aligned}$$

Since the total variation distance is bounded by 1, we then have:

$$d_{\mathrm{TV}}(\hat{\boldsymbol{q}}, \boldsymbol{q}^*) \leq 1 - \alpha = 1 - \pi^*(\mathcal{Y}^{(k)}).$$

Conclusion. The local consistency gap $\delta_{\boldsymbol{\theta},\Omega_\varepsilon}$ is defined as the maximum of this total variation distance over all choices of $\boldsymbol{y}^{(k)}$ and $S^{(k)}$ (Definition 6). Taking the maximum yields:

$$\delta_{\boldsymbol{\theta},\Omega_\varepsilon} \leq 1 - \min_{\boldsymbol{y}^{(k)},S^{(k)}} \pi^*(\mathcal{Y}^{(k)}) = \max_{\boldsymbol{y}^{(k)},S^{(k)}} \left(1 - \pi^*(\mathcal{Y}^{(k)})\right) = \max_{\boldsymbol{y}^{(k)},S^{(k)}} \boldsymbol{p}_{\Omega_\varepsilon}(\boldsymbol{s}_{\boldsymbol{\theta}})(\mathcal{Y} \setminus \mathcal{Y}^{(k)}).$$

□

B.12 Proof of Proposition 11 (Exactness of perturbed LNS on product spaces)

Proof. **Factorization of the global distribution.** Recall that the global regularized distribution is defined by the law of the perturbed maximizer $Y^* \triangleq \arg\max_{\boldsymbol{y}\in\mathcal{Y}} \langle \boldsymbol{\theta} + \varepsilon Z, \boldsymbol{y}\rangle$ with $Z \sim \mathcal{D}$. Because the feasible set is a product space $\mathcal{Y} = \prod_{m=1}^M \mathcal{Y}_m$, the dot product additively decomposes over the blocks B_m. The global maximization therefore separates into M independent local maximizations:

$$Y^* = \left(\arg\max_{\boldsymbol{y}_m \in \mathcal{Y}_m} \langle \boldsymbol{\theta}_{[B_m]} + \varepsilon Z_{[B_m]}, \boldsymbol{y}_m \rangle \right)_{m=1}^M.$$

Since the noise blocks $(Z_{[B_m]})_{m=1}^M$ are mutually independent, the components $(Y^*_{[B_m]})_{m=1}^M$ are independent random variables. Thus, the global regularized distribution $\pi^* \triangleq \boldsymbol{p}_{\Omega_\varepsilon}(\boldsymbol{s}_{\boldsymbol{\theta}})$ factorizes over the blocks:

$$\pi^*(\boldsymbol{y}) = \prod_{m=1}^M \pi^*_m(\boldsymbol{y}_{[B_m]}), \quad \text{where} \quad \pi^*_m(\boldsymbol{y}_m) \triangleq \mathbb{P}\left(\arg\max_{\boldsymbol{y}'_m \in \mathcal{Y}_m} \langle \boldsymbol{\theta}_{[B_m]} + \varepsilon Z_{[B_m]}, \boldsymbol{y}'_m \rangle = \boldsymbol{y}_m \right).$$

Conditional distribution. Let $\boldsymbol{y}^{(k)} \in \mathcal{Y}$ be a current solution, and $S^{(k)}$ be a union of blocks defining the frozen variables. Let $F^{(k)} \triangleq [d] \setminus S^{(k)}$ be the complement (the free variables), which is also a union of blocks. Due to the independence of the blocks, conditioning the global distribution π^* on the frozen variables $\boldsymbol{y}^{(k)}_{[S^{(k)}]}$ simply restricts the product to the free blocks:

$$\pi^*\left(\boldsymbol{y} \mid \boldsymbol{y}^{(k)}_{[S^{(k)}]}\right) = \prod_{B_m \subseteq F^{(k)}} \pi^*_m(\boldsymbol{y}_{[B_m]}) \quad \text{for any } \boldsymbol{y} \in \mathcal{Y}^{(k)}, \tag{18}$$

where $\mathcal{Y}^{(k)} = \{\boldsymbol{y} \in \mathcal{Y} \mid \boldsymbol{y}_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]}\}$.

Local RLNS update. By Proposition 3, the local perturbed RLNS update samples from $\boldsymbol{p}_{\Omega_\varepsilon^{\mathcal{Y}^{(k)}}}((\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]})$ by computing $\hat{Y} \triangleq \arg\max_{\boldsymbol{y}\in\mathcal{Y}^{(k)}} \langle \boldsymbol{\theta} + \varepsilon Z, \boldsymbol{y}\rangle$. Because $\boldsymbol{y}_{[S^{(k)}]}$ is fixed to $\boldsymbol{y}^{(k)}_{[S^{(k)}]}$ for all $\boldsymbol{y} \in \mathcal{Y}^{(k)}$, the maximization over $\mathcal{Y}^{(k)}$ reduces to maximizing over the free blocks:

$$\hat{Y}_{[F^{(k)}]} = \arg\max_{\boldsymbol{y}_{\text{free}} \in \prod_{B_m \subseteq F^{(k)}} \mathcal{Y}_m} \sum_{B_m \subseteq F^{(k)}} \langle \boldsymbol{\theta}_{[B_m]} + \varepsilon Z_{[B_m]}, \boldsymbol{y}_{[B_m]} \rangle.$$

This subproblem is structurally identical to the one defining the free components of the global maximizer Y^*. Since $Z \sim \mathcal{D}$, the distribution of $\hat{Y}_{[F^{(k)}]}$ is exactly $\prod_{B_m \subseteq F^{(k)}} \pi^*_m$.

Thus, the local regularized distribution matches the conditional global distribution from Eq. (18):

$$\boldsymbol{p}_{\Omega_\varepsilon^{\mathcal{Y}^{(k)}}}((\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]})(\boldsymbol{y}) = \pi^*\left(\boldsymbol{y} \mid \boldsymbol{y}^{(k)}_{[S^{(k)}]}\right).$$

By Definition 6, this exact match implies that the local consistency gap is $\delta_{\boldsymbol{\theta},\Omega_\varepsilon} = 0$. Since the local updates are equivalent to conditional sampling from π^*, Algorithm 3 precisely operates as a block Gibbs sampler targeting π^*, and its stationary distribution is exactly $\pi^{\mathrm{LNS}}_{\boldsymbol{\theta},\Omega_\varepsilon} = \boldsymbol{p}_{\Omega_\varepsilon}(\boldsymbol{s}_{\boldsymbol{\theta}})$. □

B.13 Proof of Proposition 12 (Cut-bounded consistency gap)

Proof. Consider a fixed current solution $\boldsymbol{y}^{(k)} \in \mathcal{Y}$ and a subset of frozen coordinates $S^{(k)} \subseteq [d]$, with free coordinates $F^{(k)} = [d] \setminus S^{(k)}$. Let $\hat{\boldsymbol{q}} \triangleq \boldsymbol{p}_{\Omega_\varepsilon^{\mathcal{Y}^{(k)}}}((\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]}) \in \triangle^{\mathcal{Y}^{(k)}}$ denote the local RLNS update distribution, and $\boldsymbol{q}^* \triangleq \boldsymbol{p}_{\Omega_\varepsilon}(\boldsymbol{s}_{\boldsymbol{\theta}})(\cdot \mid \boldsymbol{y}^{(k)}_{[S^{(k)}]}) \in \triangle^{\mathcal{Y}^{(k)}}$ denote the ideal global conditional distribution. Let $A \triangleq \{Y^*_{\text{relax}}(S^{(k)}) \in \mathcal{Y}\}$, so that $\overline{A} = \{Y^*_{\text{relax}}(S^{(k)}) \in W_\partial(S^{(k)})\}$. Notice that if either $\mathbb{P}(\overline{A} \mid (Y^*_{\text{relax}}(S^{(k)}))_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]}) = 1$ or $\mathbb{P}(\overline{A} \mid Y^*_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]}) = 1$, the bound holds trivially since $d_{\mathrm{TV}}(\hat{\boldsymbol{q}}, \boldsymbol{q}^*) \leq 1$; we thus assume both probabilities are strictly less than 1.

1: Reference feasible relaxed distribution. Define the relaxed local neighborhood $\mathcal{Y}^{(k)}_{\text{relax}}$ by

$$\mathcal{Y}^{(k)}_{\text{relax}} \triangleq \{\boldsymbol{y} \in \mathcal{Y}_{\text{relax}}(S^{(k)}) \mid \boldsymbol{y}_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]}\}$$

and the coupled local maximizers $\hat{Y}$ and $\hat{Y}_{\text{relax}}$ by

$$\hat{Y} \triangleq \operatorname*{arg\,max}_{\boldsymbol{y} \in \mathcal{Y}^{(k)}} \langle \boldsymbol{\theta} + \varepsilon Z, \boldsymbol{y} \rangle \sim \hat{\boldsymbol{q}} \quad \text{and} \quad \hat{Y}_{\text{relax}} \triangleq \operatorname*{arg\,max}_{\boldsymbol{y} \in \mathcal{Y}^{(k)}_{\text{relax}}} \langle \boldsymbol{\theta} + \varepsilon Z, \boldsymbol{y} \rangle.$$

Since $\mathcal{Y}_{\text{relax}}(S^{(k)}) = \mathcal{Y}_F(S^{(k)}) \times \mathcal{Y}_S(S^{(k)})$ and $Z_{[F^{(k)}]}$ is independent of $Z_{[S^{(k)}]}$, we have $(\hat{Y}_{\text{relax}})_{[F^{(k)}]} = (Y^*_{\text{relax}}(S^{(k)}))_{[F^{(k)}]}$, which is independent of $(Y^*_{\text{relax}}(S^{(k)}))_{[S^{(k)}]}$. Consequently, letting $E_\partial \triangleq \{\hat{Y}_{\text{relax}} \notin \mathcal{Y}^{(k)}\}$, we have $\mathbb{P}(E_\partial) = \mathbb{P}(\overline{A} \mid (Y^*_{\text{relax}}(S^{(k)}))_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]}) < 1$, and we can define a reference distribution $\boldsymbol{r} \in \triangle^{\mathcal{Y}^{(k)}}$ by conditioning $\hat{Y}_{\text{relax}}$ on $\overline{E_\partial}$:

$$r_{\boldsymbol{y}} \triangleq \mathbb{P}\big(\hat{Y}_{\text{relax}} = \boldsymbol{y} \,\big|\, \overline{E_\partial}\big) = \mathbb{P}\big(Y^*_{\text{relax}}(S^{(k)}) = \boldsymbol{y} \,\big|\, A,\ (Y^*_{\text{relax}}(S^{(k)}))_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]}\big) \quad \forall \boldsymbol{y} \in \mathcal{Y}^{(k)}.$$

2: Bounding the distance to the local RLNS update. Since $\mathcal{Y}^{(k)} \subseteq \mathcal{Y}^{(k)}_{\text{relax}}$, we have that on the event $\overline{E_\partial} = \{\hat{Y}_{\text{relax}} \in \mathcal{Y}^{(k)}\}$ the maximizer over $\mathcal{Y}^{(k)}_{\text{relax}}$ belongs to $\mathcal{Y}^{(k)}$, so $\hat{Y} = \hat{Y}_{\text{relax}}$ almost surely. Conditioning $\hat{\boldsymbol{q}}(\boldsymbol{y}) = \mathbb{P}(\hat{Y} = \boldsymbol{y})$ on $\overline{E_\partial}$ and E_∂ yields, for all $\boldsymbol{y} \in \mathcal{Y}^{(k)}$:

$$\hat{\boldsymbol{q}}(\boldsymbol{y}) = (1 - \mathbb{P}(E_\partial))r_{\boldsymbol{y}} + \mathbb{P}(E_\partial)\mathbb{P}(\hat{Y} = \boldsymbol{y} \mid E_\partial),$$

which implies $d_{\mathrm{TV}}(\hat{\boldsymbol{q}}, \boldsymbol{r}) \leq \mathbb{P}(E_\partial) = \mathbb{P}\big(Y^*_{\text{relax}}(S^{(k)}) \in W_\partial(S^{(k)}) \,\big|\, (Y^*_{\text{relax}}(S^{(k)}))_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]}\big)$.

3: Bounding the distance to the ideal conditional. Since $\mathcal{Y} \subseteq \mathcal{Y}_{\text{relax}}(S^{(k)})$, we have that on the event $A = \{Y^*_{\text{relax}}(S^{(k)}) \in \mathcal{Y}\}$ the equality $Y^* = Y^*_{\text{relax}}(S^{(k)})$ holds almost surely. Therefore we have that $A \cap \{Y^*_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]}\} = A \cap \{(Y^*_{\text{relax}}(S^{(k)}))_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]}\}$, and for all $\boldsymbol{y} \in \mathcal{Y}^{(k)}$, $\mathbb{P}(Y^* = \boldsymbol{y} \mid A,\, Y^*_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]}) = r_{\boldsymbol{y}}$. Conditioning $\boldsymbol{q}^*(\boldsymbol{y}) = \mathbb{P}(Y^* = \boldsymbol{y} \mid Y^*_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]})$ on A and $\overline{A}$ therefore yields:

$$\boldsymbol{q}^*(\boldsymbol{y}) = \mathbb{P}(A \mid Y^*_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]})r_{\boldsymbol{y}} + \mathbb{P}(\overline{A} \mid Y^*_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]})\mathbb{P}(Y^* = \boldsymbol{y} \mid \overline{A},\, Y^*_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]}),$$

giving $d_{\mathrm{TV}}(\boldsymbol{q}^*, \boldsymbol{r}) \leq \mathbb{P}(\overline{A} \mid Y^*_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]}) = \mathbb{P}\big(Y^*_{\text{relax}}(S^{(k)}) \in W_\partial(S^{(k)}) \,\big|\, Y^*_{[S^{(k)}]} = \boldsymbol{y}^{(k)}_{[S^{(k)}]}\big)$.

4: Triangle inequality and union bound. Combining Steps 2 and 3 via

$$d_{\mathrm{TV}}(\hat{\boldsymbol{q}}, \boldsymbol{q}^*) \leq d_{\mathrm{TV}}(\hat{\boldsymbol{q}}, \boldsymbol{r}) + d_{\mathrm{TV}}(\boldsymbol{r}, \boldsymbol{q}^*)$$

and taking the maximum over $\boldsymbol{y}^{(k)} \in \mathcal{Y}$ and $S^{(k)}$ yields the first bound on $\delta_{\boldsymbol{\theta}, \Omega_\varepsilon}$. Applying a union bound over $c \in C_\partial(S^{(k)})$ to $W_\partial(S^{(k)}) = \bigcup_{c \in C_\partial(S^{(k)})} W_c(S^{(k)})$ inside both conditional probabilities yields the second bound. □

C Experimental details for κ-hot vector estimation

This section details the experimental setup for the numerical results presented in Section 5.1.

Problem formulation. We consider the set of κ-hot vectors, $\mathcal{Y} \triangleq \{\boldsymbol{y} \in \{0,1\}^d \mid \sum_{i=1}^d y_i = \kappa\}$, with $d = 1000$ and $\kappa = 500$. The score vectors $\boldsymbol{\theta} \in \mathbb{R}^d$ are sampled from a standard Normal distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$. We run all algorithms for 10^5 iterations and average the metrics over 128 random seeds (controlling randomness of the generation of $\boldsymbol{\theta}$ as well as in the MCMC and Monte-Carlo processes) for statistical significance. To measure convergence, we compute the following mean absolute error (MAE):

$$\mathrm{MAE}^{(K)} = \frac{1}{d} \sum_{i=1}^{d} \left| \hat{\mu}_i^{(K)} - \mu_i \right|,$$

where $\hat{\boldsymbol{\mu}}^{(K)} = \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{y}^{(k)}$ is the K-th RLNS estimate of $\boldsymbol{\mu} \triangleq \mathbb{E}_{\boldsymbol{p}_\Omega(\boldsymbol{s_\theta})}[Y]$.

Variable selection. For the neighborhood system in RLNS, at each step k, we uniformly sample a random mask to freeze a fixed fraction of the variables (e.g., 70% frozen, 30% free). The local subproblem then amounts to placing exactly $\kappa - l$ ones among the free variables, where l is the number of ones already assigned in the frozen subset.

Regularization and Oracles. We evaluate the methods under two regularization regimes:

- **Shannon Regularization:** We set the temperature $\gamma = 1.0$. The local Gibbs sampling $\boldsymbol{p}_{\Omega^{\mathcal{Y}^{(k)}}}((\boldsymbol{s_\theta})_{[\mathcal{Y}^{(k)}]})$ is performed exactly using regularized dynamic programming on the free variables. The global target $\boldsymbol{\mu}$ is analytically computed using the global regularized dynamic programming formulation [Vivier-Ardisson et al., 2026b].
- **Perturbation-based Regularization:** We use standard Gaussian noise $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$ with scaling $\varepsilon = 1.0$. Local sampling is performed via perturbed local maximization, which trivially reduces to identifying the top-$(\kappa - l)$ perturbed scores among the free variables. The global target $\boldsymbol{\mu}$ is estimated to high precision via independent Monte-Carlo samples of the global perturbed MAP.

Baselines. We compare RLNS against i.i.d. Monte-Carlo sampling, which repeatedly calls the global exact regularized oracles (i.e., global perturb-and-MAP [Papandreou and Yuille, 2011, Berthet et al., 2020] for perturbation-based regularization and sampling from the global Gibbs distribution $\boldsymbol{p}_\Omega(\boldsymbol{s_\theta})(\boldsymbol{y}) \propto \exp(\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle / \gamma)$ for Shannon regularization) and Local Search MCMC [Vivier-Ardisson et al., 2026a]. For Local Search MCMC, the proposal mechanism uses the same neighborhood system as RLNS, also uniformly sampling a random mask to freeze the same fixed fraction of variables. It then uniformly samples a new valid κ-hot vector (by sampling a $(\kappa - l)$-hot vector on the free coordinates), which is then accepted or rejected via the standard Metropolis-Hastings ratio.

Every Monte-Carlo and MCMC run is performed on a CPU with 64GB of RAM and 16 cores.

D Experimental details for the generalized assignment problem

This section presents the experimental details for the numerical results presented in Section 5.2.

Instance generation. We consider GAP instances with $n = 50$ items and $m = 5$ bins. For each instance, item-bin weights w_{ij} are sampled uniformly from the discrete set $\{1, \ldots, 10\}$. To create tight constraints that force non-trivial combinatorial trade-offs, the capacity C_j for all bins in a given instance is set identically to $C = \lfloor \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij}/m^2 \rfloor$.

Dataset generation and expert policy. We generate a dataset of 1000 independent GAP instances. For each instance, we generate latent profile vectors to construct relational features:

- **Item features:** $\boldsymbol{v}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ for $i \in [n]$.
- **Bin features:** $\boldsymbol{u}_j \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ for $j \in [m]$.

The edge-wise feature vector for an item-bin pair is the concatenation of their respective profiles, $\boldsymbol{x}_{ij} = [\boldsymbol{v}_i, \boldsymbol{u}_j] \in \mathbb{R}^{20}$. To generate the ground-truth assignment labels $\boldsymbol{y} \in \{0,1\}^{n \times m}$, a randomly initialized target neural network processes the features to generate true (unseen by the model) underlying scores θ^*_{ij}, and the target assignment $\boldsymbol{y} = \hat{\boldsymbol{y}}(\boldsymbol{\theta}^*)$ is found using the `scipy.optimize.milp` global solver. A time limit of 60 seconds is enforced per instance during dataset generation.

Model architecture. We learn a mapping from the relational features $\boldsymbol{x}_{ij}$ to the predicted scores θ_{ij} using a shared multi-layer perceptron. The network operates independently on each item-bin pair and shares weights across all edges. The architecture consists of two hidden layers of size 32 with ReLU activations, followed by a linear projection to a scalar cost.

Local oracles and LNS subproblem. To scale learning, RLNS utilizes a randomized neighborhood system. At each iteration k, the assignments of all bins except one randomly selected free bin $j_k \in [m]$ are frozen. The LNS subproblem can be written as the following $0/1$ Knapsack problem:

$$\max_{\boldsymbol{y}_{[:,j_k]} \in \{0,1\}^n} \sum_{i=1}^{n} \theta_{ij_k} y_{ij_k} \quad \text{s.t.} \quad \sum_{i=1}^{n} \tilde{w}_{ij_k} y_{ij_k} \leq C,$$

where the weight $\tilde{w}_{ij_k}$ is set to $(C+1)$ if item i is already assigned to another bin to prevent assigning it twice, and to w_{ij_k} otherwise.

This subproblem admits an exact dynamic programming solution with $\mathcal{O}(nC)$ complexity [Kellerer et al., 2004]. Thus, we can easily sample from $\boldsymbol{p}_{\Omega^{\mathcal{Y}^{(k)}}_{\varepsilon}}((\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]})$ when using perturbation-based regularization by local perturbed maximization. When using Shannon regularization, one can sample from the local Gibbs distribution $\boldsymbol{p}_{\Omega^{\mathcal{Y}^{(k)}}}((\boldsymbol{s}_{\boldsymbol{\theta}})_{[\mathcal{Y}^{(k)}]})$ via regularized dynamic programming with the same complexity using Vivier-Ardisson et al. [2026b, Algorithm 3].

Training details. For Fig. 2a, all models are trained for $10,000$ steps using the Adam optimizer with a learning rate of 2×10^{-3} and a batch size of 32, on a CPU with 64GB of RAM and 16 cores. For perturbation-based regularization (in both RLNS and PFY), we use standard Gaussian noise $\mathcal{D} = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ with temperature $\varepsilon = 1.0$.

Gradient variance analysis. To robustly evaluate the stability of the RLNS stochastic gradients in Fig. 2b (top), we measure the trace of the gradient covariance matrix using repeated sampling. For these experiments, we fix a single GAP instance and sample a batch of 64 logit vectors $\boldsymbol{\theta}$ from a standard Normal distribution. For each score vector, we draw 50 independent gradient samples to compute the per-instance covariance trace. The reported variance metrics represent the average across this batch of 64 queried score vectors. All error bars indicate the 95% confidence intervals of this mean across the batch, calculated using the standard normal multiplier (1.96). Explicit random seeds are used to strictly control the generation of the underlying problem instance weights, the queried score vectors, and the internal stochasticity of the MCMC and perturbation sampling mechanisms.

Oracle execution time benchmarking. To ensure statistically significant time comparisons between the local dynamic programming oracle and the global MILP solver in Fig. 2b (bottom), we benchmark execution times across 100 independent trials for each problem scale ($(n, m) = (100, 10)$, $(n, m) = (200, 20)$, and $(n, m) = (400, 40)$). Within each scale, the underlying item weights and bin capacities are generated using a fixed random seed (following the same generation process as explained above). For each of the 100 trials, we sample a new logit vector $\boldsymbol{\theta}$ and, for the local oracle, uniformly sample a new free bin to re-optimize. The reported execution times are the arithmetic means over the 100 trials, with error bars representing the 95% confidence intervals derived from a Student's t-distribution.

E Experimental details for stochastic vehicle scheduling

This section details the complete experimental setup, dataset generation procedure, model architecture, and hyperparameters for the stochastic vehicle scheduling problem (SVSP) experiments presented in Section 5.3. While the problem formulation and dataset generation process are standard and identical to, e.g., Parmentier [2022], Dalle et al. [2022], Hoppe et al. [2025], we detail them here for completeness.

E.1 Problem formulation

The problem is modeled on a directed acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = n + 2$ nodes (a source depot 0, n job nodes $[n]$, and a sink depot $n + 1$) and $|\mathcal{E}|$ edges. An edge $(i, j) \in \mathcal{E}$ exists from source to every job, from every job to sink, and from job i to job j if the deterministic end time of task i plus travel time allows reaching task j before its start. The number of edges varies across instances, as it depends on the spatial and temporal configuration of the tasks.

A feasible routing $\boldsymbol{y} \in \{0, 1\}^{|\mathcal{E}|}$ satisfies flow conservation and unit demand constraints at every job node:

$$\sum_{e \in \delta^-(v)} y_e = 1 \quad \text{and} \quad \sum_{e \in \delta^+(v)} y_e = 1 \quad \forall v \in [n],$$

where $\delta^-(v)$ and $\delta^+(v)$ denote the sets of incoming and outgoing edges of v. Each vehicle corresponds to a path from source to sink, so the number of vehicles is $\sum_{e \in \delta^+(0)} y_e$.

Given a routing $\boldsymbol{y}$ and a scenario $\omega \in [S]$, delays propagate along each vehicle path via the recursion $R_v^{(\omega)} = \varepsilon_v^{(\omega)} + \max(R_u^{(\omega)} - \sigma_{uv}^{(\omega)}, 0)$, where $\varepsilon_v^{(\omega)}$ is the intrinsic delay at node v, $\sigma_{uv}^{(\omega)}$ is the slack (time margin) on edge (u, v), and (u, v) is the predecessor edge of v in the routing. The total cost of routing $\boldsymbol{y}$ is the sum of the vehicle cost and the average total delay cost over the scenarios:

$$c(\boldsymbol{y}) = c_{\text{vehicle}} \cdot \sum_{e \in \delta^+(0)} y_e + c_{\text{delay}} \cdot \frac{1}{S} \sum_{\omega=1}^{S} \sum_{v=1}^{n} R_v^{(\omega)}.$$

E.2 Dataset generation and expert policy

We generate a dataset of 1000 independent SVSP instances. For each instance, we sample a random city layout simulating spatial areas and transit delays.

- **Stochasticity:** Task locations, start times, and travel times are sampled randomly. Intrinsic task delays ($\varepsilon_v^{(\omega)}$) and regional inter-area travel delays follow LogNormal distributions, generating the 10 distinct delay scenarios per instance.
- **Expert policy:** The ground-truth solution $\boldsymbol{y}$ for each instance is computed by solving a two-stage stochastic MILP that minimizes the expected total cost $c(\boldsymbol{y})$ with full knowledge of all S scenarios. To handle the bilinear delay-propagation term $y_{uv} R_u^{(\omega)}$ without requiring a big-M parameter, we use exact McCormick envelope linearizations. The MILP is solved using SciPy with a time limit of 120 seconds per instance.
- **Size:** Each instance features $n = 50$ job tasks and $S = 10$ stochastic scenarios, which is the largest problem size handled by the expert solver. In terms of industrial applications, this scale corresponds to standard long-haul fleet aircraft routing problems.

E.3 Features and model architecture

For each edge $e \in \mathcal{E}$ in a given graph, we construct a 20-dimensional feature vector $\boldsymbol{x}_e \in \mathbb{R}^{20}$. The features capture the deterministic and stochastic characteristics of the edge:

- **Feature 1:** The deterministic travel time for the edge.
- **Feature 2:** The vehicle cost (equal to c_{vehicle} if the edge originates from the source depot, and 0 otherwise).
- **Features 3–11:** The 9 deciles of the slack (time margin) distribution $\sigma_{uv}^{(\omega)}$ computed across the $S = 10$ scenarios.
- **Features 12–20:** The cumulative distribution function (CDF) of the slack distribution evaluated at 9 predefined time thresholds: $\{-100, -50, -20, -10, 0, 10, 50, 200, 500\}$.

The predictive model uses a standard feed-forward architecture. It applies a shared two-layer perceptron with a hidden dimension of 64 independently to each edge's feature vector $\boldsymbol{x}_e$ to output the predicted scalar cost θ_e.

Given predicted edge costs $\boldsymbol{\theta} \in \mathbb{R}^{|\mathcal{E}|}$, the combinatorial optimization layer is then a flow linear program (LP) $\hat{\boldsymbol{y}}(\boldsymbol{\theta}) \triangleq \arg\max_{\boldsymbol{y} \in \mathcal{Y}} \langle \boldsymbol{\theta}, \boldsymbol{y} \rangle$.

E.4 Training details and hyperparameters

All models are trained for $10,000$ steps using the Adam optimizer with a learning rate of 10^{-4} and a batch size of 32. The dataset is split into 70% training, 15% validation, and 15% testing. All training configurations are repeated across 10 different random seeds controlling dataset and model initialization randomness for statistical significance. Each training is performed on a CPU with 64GB of RAM and 16 cores.

RLNS-specific parameters. Our proposed method is parameterized as follows:

- **Iterations:** We run $K = 1$ RLNS iteration per training step to compute the gradient estimator.
- **Regularization:** We use perturbation-based regularization with $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ standard Gaussian noise and a temperature scaling parameter $\varepsilon = 1.0$.
- **Initialization:** RLNS is initialized directly at the ground-truth expert solution.
- **Variable selection scheme:** We use a route-based mask generation heuristic. At each step, we identify the distinct vehicle routes present in the current routing assignment. We select 66% of these routes uniformly at random and freeze all variables corresponding to the tasks visited by those routes.

Baseline parameters. The baselines share the identical underlying model architecture, optimization hyperparameters, and evaluation metrics:

- **Global perturbed Fenchel-Young loss (PFY) [Berthet et al., 2020]:** Uses the same perturbation-based regularization setup (Gaussian noise, $\varepsilon = 1.0$), but calls the exact global MAP oracle over the full graph $\mathcal{G}$ at every training step.
- **Blackbox differentiation (DBB) [Vlastelica et al., 2020]:** Continuous interpolation of the solver. We performed a hyperparameter sweep over the interpolation strength $\lambda \in \{10^{-1}, 5 \times 10^{-1}, 1, 5, 10^1, 10^2, 5 \times 10^2, 10^3, 5 \times 10^3, 10^4\}$ and report the best performing configuration, which was $\lambda = 10^2$.
- **Binary cross-entropy (BCE):** An unstructured binary cross-entropy loss applied directly to the edge-level predictions $\boldsymbol{\theta}$ against the expert binary labels $\boldsymbol{y}$. See, e.g., Joshi et al. [2019] for a similar approach on a routing combinatorial optimization problem.
- **Greedy policy:** A non-learned baseline that solves a standard minimum-cost flow problem on the DAG, placing negative costs on the source edges to minimize the total number of vehicles, entirely ignoring the stochastic delays.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our abstract and introduction clearly state the claims and contributions of the paper (Section 1).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in a dedicated paragraph in the Conclusion section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All proofs are included in the appendix, and provide the full set of needed assumptions (Appendix B).

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The main practical contributions of the paper are Algorithms 2 and 3, which are described in detail. The numerical experiments are carefully documented in Section 5 and Appendices C to E to ensure they are as reproducible as possible.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: All data used is either public or synthetic, with a clearly described generation process.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We present the experimental settings with sufficient level of detail to fully appreciate the results in the main text, and give full details in the appendix (Appendices C to E).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We average experiments over multiple runs with varying seeds and report error bars where relevant in consequence, with statistical significance statements and precise details in appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We include sufficient relevant information on the compute resources.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have carefully reviewed the NeurIPS Code of Ethics and ensured that all aspects of the research comply with its guidelines.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: As this is a foundational research paper, we do not foresee direct negative societal impacts in its current form. Potential negative impacts would depend on downstream applications, which are beyond the scope of this work.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This work does not pose such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification: We do not use existing assets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We do not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.